

IBM Rational Developer for System z
バージョン 9.1.1

構成ガイド



IBM Rational Developer for System z
バージョン 9.1.1

構成ガイド



— お願い —

本書をご使用になる前に、247 ページの『特記事項』に記載されている情報をお読みください。

本書は、IBM Rational Developer for System z バージョン 9.1.1 (プログラム番号 5724-T07) および新しい版で明記されていない限り、以降のすべてのリリースおよびモディフィケーションに適用されます。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

お客様の環境によっては、資料中の円記号がバックスラッシュと表示されたり、バックスラッシュが円記号と表示されたりする場合があります。

原典： SC23-7658-13

IBM Rational Developer for System z

Version 9.1.1

Configuration Guide

Fourteenth edition (December 2014)

発行： 日本アイ・ビー・エム株式会社

担当： トランスレーション・サービス・センター

© Copyright IBM Corporation 2000, 2014.

目次

図	vii
---	-----

表	ix
---	----

本書について	xi
--------	----

本書の対象読者	xii
変更の要約	xii
文書内容の説明	xvi
計画	xvi
基本的なカスタマイズ	xvi
(オプション) 共通アクセス・リポジトリ・マネージャー (CARMA)	xvi
(オプション) SCLM Developer Toolkit	xvii
(オプション) Application Deployment Manager (非推奨)	xvii
(オプション) ホスト・ベースのコード分析	xvii
(オプション) その他のカスタマイズ・タスク	xviii
インストール検査	xviii
セキュリティ定義	xviii
マイグレーション・ガイド	xix
オペレーター・コマンド	xix
ホスト構成リファレンス	xix

IBM Rational Developer for System

z ホスト構成ガイド	1
------------	---

第 1 章 計画	3
----------	---

マイグレーションに関する考慮事項	3
計画に関する考慮事項	3
製品の概要	3
スキルの要件	4
時間の要件	4
インストール前の考慮事項	4
インストール・ユーザー ID	5
必要な製品	5
必要なリソース	6
構成前の考慮事項	11
ワークロード管理	11
リソース使用量とシステム限度	11
必要な製品の必須の構成	11
ユーザー ID に関する考慮事項	12
サーバーに関する考慮事項	13
構成方法	14
デプロイメント前の考慮事項	14
クライアント・チェックリスト	15

第 2 章 基本的なカスタマイズ	19
------------------	----

要件およびチェックリスト	19
カスタマイズのセットアップ	20
PARMLIB の変更	21

BPXPRMxx での z/OS UNIX 限度の設定	21
COMMNDxx への開始タスクの追加	23
IEASVCxx での SVC 定義	23
LPALSTxx での LPA 定義	24
PROGxx での APF 許可	25
PROGxx での LINKLIST 定義	25
必要な LINKLIST 定義と LPA 定義	27
他の製品用の LINKLIST 定義	29
PROCLIB の変更	29
JMON、JES ジョブ・モニター開始タスク	29
DBGMR、デバッグ・マネージャー開始タスク	30
RSED、RSE デーモン開始タスク	31
PARM 変数の JCL での制限	32
TMPDIR 処理	33
STDENV 処理	34
ELAXF* リモート・ビルド・プロシージャ	35
セキュリティ定義	37
FEJJCNFG、JES ジョブ・モニター構成ファイル	38
rsed.envvars、RSE 構成ファイル	43
RSE サーバーに使用可能な PORTRANGE の定義	55
_RSE_JAVAOPTS での追加 Java 始動パラメータの定義	56
_RSE_ISPF_OPTS での追加 Java 始動パラメータの定義	66
ISPF.conf、ISPF の TSO/ISPF クライアント・ゲートウェイ構成ファイル	67
オプションのコンポーネント	68
インストール検査	69

第 3 章 (オプション) 共通アクセス・リポジトリ・マネージャー (CARMA)	71
---	----

要件およびチェックリスト	71
サーバー始動方式とアクティブ RAM の選択	72
CARMA サーバーの始動	72
CRASTART	72
バッチ実行依頼	72
(非推奨) TSO/ISPF クライアント・ゲートウェイ	73
実動 RAM	73
CA Endeavor SCM RAM	73
CA Endeavor SCM パッケージ RAM	73
サンプル RAM	73
PDS RAM	73
スケルトン RAM	73
SCLM RAM	74
事前構成済みの RAM とサーバー始動の組み合わせ	74
CA Endeavor SCM RAM による CRASTART	74
CARMA VSAM データ・セットの作成	74
CRASRV.properties のカスタマイズ	75
crastart.endevor.conf のカスタマイズ	75

(オプション) 追加の CA Endeavor SCM RAM の カスタマイズ	76
サンプル RAM による CRASTART	77
CARMA VSAM データ・セットの作成	77
CARMA	77
サンプル RAM	77
CRASRV.properties のカスタマイズ	78
crastart.conf のカスタマイズ	78
(オプション) 追加のカスタム RAM のカスタマイ ズ	79
CA Endeavor SCM RAM によるバッチ実行依頼	79
CARMA VSAM データ・セットの作成	80
CRASRV.properties のカスタマイズ	80
CRASUBCA のカスタマイズ	80
(オプション) 追加の CA Endeavor SCM RAM の カスタマイズ	82
サンプル RAM によるバッチ実行依頼	82
VSAM データ・セットの作成	82
CARMA	82
サンプル RAM	82
CRASRV.properties のカスタマイズ	83
CRASUBMT のカスタマイズ	83
(オプション) 追加のカスタム RAM のカスタマイ ズ	84
CARMA 構成詳細	85
CRASRV.properties、CARMA への RSE インター フェース	85
crastart*.conf、CRASTART サーバーの始動	88
CRASTART ログ・ファイルの収集	91
CRASUB*、バッチ実行依頼サーバー始動	92
CARMA VSAM データ・セット	94
CRADEF、構成データ・セット	94
CRAMSG、メッセージ・データ・セット	94
CRASTRS、カスタム・ストリング・データ・ セット	94
CARMA VSAM のマイグレーションに関する 注	95
CARMA Repository Access Manager (RAM)	95
CA Endeavor SCM RAM	96
CA Endeavor SCM パッケージ RAM	96
PDS RAM	96
スケルトン RAM	97
SCLM RAM	97
CRACFG、CRASHOW、および CRATMAP、CA Endeavor SCM RAM 構成ファイル	98
CRACFG、CA Endeavor SCM RAM と SCM の相互作用	98
CRASHOW、CA Endeavor SCM RAM のデフォ ルト・フィルター	98
CRATMAP、CA Endeavor SCM RAM のファ イル拡張子のマッピング	98
CRANDVRA、CA Endeavor SCM RAM 割り振り exec	99
CA Endeavor SCM RAM バッチ・アクション	100
CRABCFG、CA Endeavor SCM RAM バッ チ・アクション構成	101

CRABATCA、CA Endeavor SCM RAM バッ チ・アクション JCL	102
CRABJOBC、CA Endeavor SCM RAM バッ チ・アクション JOB カード	103
CRAALLOC、カスタム RAM 割り振り exec	103
CARMA 戻りコード	104
(オプション) 複数の RAM のサポート	104
例	104
(オプション) カスタム割り振り exec	106
(オプション) CARMA ユーザー出口	107
(オプション) IRXJCL と CRAXJCL	107
CRAXJCL の作成	108

第 4 章 (オプション) SCLM Developer

Toolkit 109

要件およびチェックリスト	109
前提条件	110
SCLMDT 用の ISPF.conf の更新	110
SCLMDT 用の rsed.envvars の更新	111
(オプション) ロング/ショート・ネーム変換	112
LSTRANS.FILE (ロング/ショート・ネーム変換 VSAM) の作成	112
ロング/ショート・ネーム変換用の rsed.envvars の更新	114
(オプション) Ant のインストールおよびカスタマイ ズ	114
SCLMDT 用の SCLM の更新	115
WORKAREA および /tmp からの古いファイルの除 去	116

第 5 章 (オプション) Application

Deployment Manager (非推奨) 117

要件およびチェックリスト	117
CRD リポジトリ	118
CICS 管理ユーティリティ	119
RESTful と Web サービス	119
RESTful インターフェースを使用する CRD サーバ ー	120
CICS 主接続領域	120
CICS 非主接続領域	120
(オプション) CRD サーバー・トランザクション ID のカスタマイズ	121
Web サービス・インターフェースを使用する CRD サーバー	121
パイプライン・メッセージ・ハンドラー	122
CICS 主接続領域	123
CICS 非主接続領域	123
(オプション) マニフェスト・リポジトリ	124

第 6 章 (オプション) ホスト・ベースの

コード分析 125

要件およびチェックリスト	125
コード・レビュー	125
コード・レビュー・プロセスの変更	126
コード・カバレッジ	126

コード・カバレッジの単一の呼び出し	126
コード・カバレッジの複数の呼び出し	127
コード・カバレッジの出力	127
第 7 章 (オプション) その他のカスタマイズ・タスク	129
(オプション) pushtoclient.properties、ホスト・ベースのクライアント制御	129
(オプション) ssl.properties、RSE SSL 暗号化	133
(オプション) rsecomm.properties、RSE トレース	135
(オプション) include.conf、C/C++ コンテンツ・アシスト用の強制インクルード	137
(オプション) z/OS UNIX サブプロジェクト	138
REXEC または SSH のセットアップ	139
(オプション) インクルード・プリプロセッサのサポート	139
(オプション) Enterprise COBOL および PL/I での xUnit サポート	140
(オプション) エンタープライズ・サービス・ツール・サポート	141
(オプション) CICS 双方向言語サポート	142
(オプション) 生成コードの診断用 IRZ メッセージ	143
(オプション) 統合デバッガー	144
統合デバッガーと COBOL v4	146
統合デバッガーおよびその他の言語環境プログラム・ベースのデバッガー	146
統合デバッガーと暗号化通信	146
統合デバッガー構成パラメーター	147
統合デバッガーの parmlib 更新	147
統合デバッガーの TCP/IP 更新	148
統合デバッガーのセキュリティ更新	148
統合デバッガーの言語環境プログラム (Language Environment) の更新	149
統合デバッガーの CICS アップデート	150
統合デバッガーの IMS の更新	151
統合デバッガーの DB2 ストアード・プロシージャの更新	151
統合デバッガーの ELAXF* の更新	151
(オプション) 問題判別ツールのサポート	152
(オプション) DB2 および IMS のデバッグのサポート	152
(オプション) File Manager のサポート	153
(オプション) WORKAREA と /tmp のクリーンアップ	154
第 8 章 インストール検査	155
開始タスクの検査	155
JMON、JES ジョブ・モニター	155
RSED、RSE デーモン	155
DBGMR、デバッグ・マネージャー	159
IVP オペレーター・コマンド	160
PassTicket の再使用可能性	160
RSE デーモン接続	160
ISPF クライアント・ゲートウェイ	160
サービスの検査	162
IVP の初期化	162

ポート可用性	163
TCP/IP のセットアップ	164
RSE デーモン接続	165
JES ジョブ・モニター接続	165
ISPF の TSO/ISPF クライアント・ゲートウェイ接続	166
(オプション) CARMA 接続	167
(オプション) SCLMDT 接続	168

第 9 章 セキュリティー定義	171
要件およびチェックリスト	171
セキュリティの設定およびクラスをアクティブにする	173
Developer for System z ユーザーの OMVS セグメントを定義する	174
Developer for System z 開始タスクの定義	174
セキュアな z/OS UNIX サーバーとして RSE を定義する	176
RSE の MVS プログラム制御ライブラリーを定義する	176
RSE の PassTicket サポートの定義	177
RSE の z/OS UNIX ファイル・アクセス許可の定義	178
RSE のアプリケーション保護の定義	179
RSE の z/OS UNIX プログラム制御ファイルを定義する	179
JES コマンド・セキュリティを定義する	180
統合デバッガーへのアクセスの定義	182
データ・セット・プロファイルを定義する	182
セキュリティー設定の検査	186

第 10 章 マイグレーション・ガイド	189
マイグレーションに関する考慮事項	189
前に構成したファイルのバックアップ	189
バージョン 9.1 のマイグレーションに関する注意事項	190
IBM Rational Developer for System z、FMID HHOP910	190
IBM Rational Developer for System z Host Utilities、FMID HAKG910	191
バージョン 9.0 からバージョン 9.1 へのマイグレーション	191
IBM Rational Developer for System z、FMID HHOP910	192
構成可能なファイル	194
IBM Rational Developer for System z Host Utilities、FMID HAKG910	201
構成可能なファイル	201
バージョン 8.5 からバージョン 9.0 へのマイグレーション	201
IBM Rational Developer for System z、FMID HHOP900	202
構成可能なファイル	204
IBM Rational Developer for System z Host Utilities、FMID HAKG900	210
構成可能なファイル	210
バージョン 8.5 のマイグレーションに関する注	211

バージョン 8.0.1 からバージョン 8.5 へのマイグレーション	213
IBM Rational Developer for System z、FMID HHOP850	213
構成可能なファイル	216

第 11 章 オペレーター・コマンド . . . 221

Start (S)	221
統合デバッガー	221
JES ジョブ・モニター	222
RSE デーモン	222
Modify (F)	223
統合デバッガー	223
JES ジョブ・モニター	225
RSE デーモン	227
Stop (P)	238
構文図の読み方	238
記号	238
オペランド	238
構文例	239
非英数字およびブランク・スペース	239
複数のオペランドの選択	239
1 行より長い場合	239
構文フラグメント	239

ホスト構成リファレンス	239
Developer for System z について	240
セキュリティーに関する考慮事項	240
TCP/IP に関する考慮事項	240
WLM に関する考慮事項	240
チューニングに関する考慮事項	240
パフォーマンスに関する考慮事項	240
クライアントへのプッシュ機能に関する考慮事項	241
CICSTS に関する考慮事項	241
ユーザー出口に関する考慮事項	241
TSO 環境のカスタマイズ	241
複数のインスタンスの実行	241
構成問題のトラブルシューティング	241
SSL および X.509 認証のセットアップ	242
AT-TLS のセットアップ	242
TCP/IP のセットアップ	242

参考文献 243

参考資料	243
情報資料	246

特記事項 247

著作権使用許諾	250
商標の帰属表示	250



1. JMON: JES ジョブ・モニター開始タスク	30	25. CRASUB*: バッチ実行依頼を使用した	
2. DBGMR: デバッグ・マネージャー開始タスク	31	CARMA の始動	93
3. RSED: RSE デーモン開始タスク	31	26. CRACFG - CA Endeavor SCM RAM と SCM	
4. RSED - 代替の RSE デーモンの始動	33	の相互作用	98
5. rsed.stdin.sh: 代替 RSE デーモンの始動	33	27. CRASHOW - CA Endeavor SCM RAM のデフ	
6. RSED: 代替 TMPDIR 処理	34	ォルト・フィルター	98
7. rsed.stdeny: 代替 TMPDIR 処理	34	28. CRATMAP: CA Endeavor SCM RAM のデフ	
8. RSED: STDENV 処理	35	ォルト・フィルター	99
9. rsed.envvars: STDENV 処理	35	29. CRABCFG: CA Endeavor SCM RAM バッチ・	
10. FEJCNFG、JES ジョブ・モニター構成ファイ		アクション構成	101
ル	39	30. CRABATCA: CA Endeavor SCM RAM バッ	
11. rsed.envvars: RSE 構成ファイル	45	チ・アクション JCL	102
12. rsed.envvars: RSE 構成ファイル (続き)	46	31. CRABJOB: CA Endeavor SCM RAM バッ	
13. rsed.envvars: RSE 構成ファイル (続き)	47	チ・アクション JOB カード	103
14. ISPF.conf: ISPF 構成ファイル	67	32. SCLMDT 用の ISPF.conf の更新	111
15. CRASRV.properties: CA Endeavor SCM RAM に		33. SCLMDT 用の rsed.envvars の更新	112
よる CRASTART	75	34. FLM02LST: ロング/ショート・ネーム変換セッ	
16. crastart.endevor.conf: CA Endeavor SCM RAM に		トアップ JCL	113
よる CRASTART	76	35. pushtoclient.properties: ホスト・ベースのクラ	
17. CRASRV.properties: サンプル RAM による		イアント制御構成ファイル	130
CRASTART	78	36. ssl.properties - SSL 構成ファイル	134
18. crastart.conf: サンプル RAM による		37. rsecomm.properties - ロギング構成ファイル	136
CRASTART	79	38. include.conf - C/C++ コンテンツ・アシスト用	
19. CRASRV.properties - CA Endeavor SCM RAM		の強制インクルード	138
によるバッチ実行依頼	80	39. START DBGMR オペレーター・コマンド	221
20. CRASUBCA: CA Endeavor SCM RAM によるバ		40. START JMON オペレーター・コマンド	222
ッチ実行依頼	81	41. START RSED オペレーター・コマンド	222
21. CRASRV.properties: サンプル RAM によるバッ		42. MODIFY DBGMR オペレーター・コマンド	223
チ実行依頼	83	43. MODIFY JMON オペレーター・コマンド	225
22. CRASUBMT: サンプル RAM によるバッチ実		44. MODIFY RSED オペレーター・コマンド	227
行依頼	84	45. MODIFY RSED オペレーター・コマンド (続	
23. CRASRV.properties - CARMA 構成ファイル	85	き)	228
24. crastart*.conf: CRASTART を使用した CARMA		46. STOP オペレーター・コマンド	238
サーバーの始動	90		

表

1. 必要なリソース	6	17. クライアントへのプッシュのグループ・サポ ート	132
2. オプションのリソース	7	18. SSL 証明書の保管メカニズム	133
3. 必須タスクに必要な管理者	8	19. 有効な鍵ストアのタイプ	135
4. オプションのタスクに必要な管理者	9	20. サービス用の IVP	162
5. クライアント・チェックリスト: 必須部分	15	21. セキュリティー・セットアップの変動要素	171
6. クライアント・チェックリスト: オプション部 分	16	22. JES2 ジョブ・モニターのオペレーター・コマ ンド	181
7. ロード・モジュールと機能のマッチング	26	23. JES3 ジョブ・モニターのオペレーター・コマ ンド	181
8. サンプル ELAXF* プロシージャ	35	24. バージョン 9.1.0 のカスタマイズ	195
9. ELAXF* 高位修飾子チェックリスト	36	25. Host Utilities バージョン 9.0 のカスタマイズ	201
10. ELAXF*	37	26. バージョン 9.0 のカスタマイズ	204
11. LIMIT_COMMANDS コマンドの許可のマトリ ックス	41	27. Host Utilities バージョン 9.0 のカスタマイズ	211
12. デバッグ・マネージャーへの自動再接続	64	28. バージョン 8.5 のカスタマイズ	217
13. CARMA 戻りコード	104	29. スレッド・プールのエラー状況	230
14. SCLM 管理者チェックリスト	116	30. 参考資料	243
15. デフォルトの CRD サーバー・トランザクシ ョン ID	121	31. 参照される Web サイト	245
16. デフォルトの CRD サーバー・トランザクシ ョン ID	122	32. 情報資料	246

本書について

本書では、IBM® Rational® Developer for System z® の機能の構成について説明しています。ここには、ご使用の z/OS® ホスト・システム上に IBM Rational Developer for System z バージョン 9.1.1 を構成するための説明が記載されています。

これ以降、本書では以下の名前が使用されています。

- *IBM Rational Developer for System z* は *Developer for System z* と呼ばれます。
- *IBM Rational Developer for System z* 統合デバッガーは統合デバッガーと呼ばれます。
- 共通アクセス・リポジトリ・マネージャー は、*CARMA* と省略されます。
- *Software Configuration and Library Manager Developer Toolkit* は *SCLM Developer Toolkit* と呼ばれ、*SCLMDT* と省略されます。
- *IBM z/OS Automated Unit Testing Framework* は *zUnit* と呼ばれます。
- *z/OS UNIX* システム・サービス は、*z/OS UNIX* と呼ばれます。
- 顧客情報管理システム (*CICS*) *Transaction Server* は、*CICSTS* と呼ばれ、*CICS*® と略されます。

本書は、Developer for System z のホスト・システム構成を説明した文書セットの一部です。これらの文書は、それぞれ特定の読者を対象としています。Developer for System z の構成を行うために、すべての資料に目を通す必要はありません。

- 「*Rational Developer for System z* ホスト構成ガイド」(SC88-5663) は、すべての計画タスク、構成タスク、およびオプション (任意指定のものを含む) について説明し、代替シナリオを提供しています。
- 「*Rational Developer for System z* ホスト構成リファレンス」(SA88-4226) は、Developer for System z の設計について説明し、Developer for System z、z/OS コンポーネント、および Developer for System z に関連するその他の製品 (WLM および CICS など) のさまざまな構成タスクに関する背景情報を提供しています。
- 「*Rational Developer for System z* ホスト構成クイック・スタート・ガイド」(GI88-4171) は、Developer for System z の最小限のセットアップについて説明しています。
- 「*Rational Developer for System z* ホスト構成ユーティリティー・ガイド」(SA88-4197) は、ホスト構成ユーティリティー (Developer for System z の基本的かつ共通したオプションのカスタマイズ・ステップを紹介する ISPF パネル・アプリケーション) について説明しています。

本書の情報は、すべての IBM Rational Developer for System z バージョン 9.1.1 パッケージに適用されます。

本書の最新バージョンについては、「*IBM Rational Developer for System z* 構成ガイド」(SC88-5663) (<http://www-05.ibm.com/e-business/linkweb/publications/servlet/pbi.wss?CTY=US&FNC=SRX&PBL=SC88-5663>) を参照してください。

インストールの説明、ホワイト・ペーパー、ポッドキャスト、およびチュートリアルを含む、資料一式の最新バージョンについては、IBM Rational Developer for System z Web サイトのライブラリー・ページ (http://www-01.ibm.com/software/sw-library/en_US/products/Z964267S85716U24/) を参照してください。

本書の対象読者

本書は、IBM Rational Developer for System z バージョン 9.1.1 のインストールおよび構成を行うシステム・プログラマーを対象にしています。

本書では、製品を完全にセットアップするために必要な手順を、デフォルト以外のシナリオも含め、詳細に記載しています。構成の計画と実行に役立つ基本的な情報については、「*IBM Rational Developer for System z* ホスト構成リファレンス」(SA88-4226) に記載されています。本書を使用するには、z/OS UNIX システム・サービスおよび MVS™ ホスト・システムに精通している必要があります。

変更の要約

ここでは、「*IBM Rational Developer for System z* バージョン 9.1.1 ホスト構成ガイド」(SC88-5663-13) (2014 年 12 月に改訂) の変更点を要約します。

本文または図表に対して技術的な変更または追加が行われている場合には、その個所の左側に縦線を引いて示してあります。

新しい情報:

- バージョン 9.1.1 でのマイグレーションに関する情報。190 ページの『バージョン 9.1 のマイグレーションに関する注意事項』を参照してください。
- DBGMR 開始タスク定義の削除されたオプション。30 ページの『DBGMR、デバッグ・マネージャー開始タスク』を参照してください。
- 統合デバッガーの新しいセキュリティー・セットアップ。182 ページの『統合デバッガーへのアクセスの定義』を参照してください。
- Modify Display Process コマンドでの RSE スレッド・プールの新しい rejectLogon 状況。221 ページの『第 11 章 オペレーター・コマンド』を参照してください。
- rsed.envvars 内の新しいディレクティブおよび除去されたディレクティブ。43 ページの『rsed.envvars、RSE 構成ファイル』を参照してください。
- オプションの統合デバッガーに関する変更されたセットアップ手順 144 ページの『(オプション) 統合デバッガー』を参照してください。

本書には、「*IBM Rational Developer for System z* バージョン 9.1 ホスト構成ガイド」(SC88-5663-12) に記載されていた情報が含まれています。

新しい情報:

- バージョン 9.1.0 でのマイグレーション情報。191 ページの『バージョン 9.0 からバージョン 9.1 へのマイグレーション』を参照してください。
- RSED 開始タスク定義の新しいオプション。34 ページの『STDENV 処理』を参照してください。

- rsed.envvars 内の新しいディレクティブおよび変更されたディレクティブ。43 ページの『rsed.envvars、RSE 構成ファイル』を参照してください。
- 新しいオペレーター・コマンド。221 ページの『第 11 章 オペレーター・コマンド』を参照してください。
- 統合デバッガーの新しいオプション。144 ページの『(オプション) 統合デバッガー』を参照してください。

本書には、「*IBM Rational Developer for System z* バージョン 9.0.1 ホスト構成ガイド」(SC88-5663-09)に記載されていた情報が含まれています。

新しい情報:

- オプションの統合デバッガーに関する追加のセットアップ手順 144 ページの『(オプション) 統合デバッガー』を参照してください。

本書には、「*IBM Rational Developer for System z* バージョン 9.0.1 ホスト構成ガイド」(SC88-5663-08)に記載されていた情報が含まれています。

新しい情報:

- バージョン 9.0.1 でのマイグレーション情報。
- 新しいオプションの PARMLIB 更新。21 ページの『PARMLIB の変更』を参照してください。
- 新しいオプションの開始タスク。29 ページの『PROCLIB の変更』を参照してください。
- 新しいオペレーター・コマンド。221 ページの『第 11 章 オペレーター・コマンド』を参照してください。
- rsed.envvars の新しいディレクティブおよび変更されたディレクティブ。43 ページの『rsed.envvars、RSE 構成ファイル』を参照してください。
- CRASRV.properties 内の新しいディレクティブ。85 ページの『CRASRV.properties、CARMA への RSE インターフェース』を参照してください。
- CARMA CA Endeavor® SCM RAM 用の新しい構成ファイル。98 ページの『CRACFG、CRASHOW、および CRATMAP、CA Endeavor® SCM RAM 構成ファイル』を参照してください。
- CARMA の始動時に呼び出されるユーザー出口のサポート。107 ページの『(オプション) CARMA ユーザー出口』を参照してください。

本書には、「*IBM Rational Developer for System z* バージョン 9.0 ホスト構成ガイド」(SC88-5663-07)に記載されていた情報が含まれています。

新しい情報:

- バージョン 9.0 でのマイグレーション情報。201 ページの『バージョン 8.5 からバージョン 9.0 へのマイグレーション』を参照してください。
- rsed.envvars の新しいディレクティブおよび変更されたディレクティブ。43 ページの『rsed.envvars、RSE 構成ファイル』を参照してください。
- FEJJCNFG の新しいディレクティブおよび削除されたディレクティブ。38 ページの『FEJJCNFG、JES ジョブ・モニター構成ファイル』を参照してください。

- 新しい ELAXF* PROCLIB メンバー。35 ページの『ELAXF* リモート・ビルド・プロシージャ』を参照してください。
- 新しい JMON オペレーター・コマンドおよび RSED オペレーター・コマンド。223 ページの『Modify (F)』を参照してください。
- ホスト・ベースのコード分析に関する追加情報。125 ページの『第 6 章 (オプション) ホスト・ベースのコード分析』を参照してください。

削除された情報:

- LOCKD 開始タスクが使用されなくなったため、ロック・デーモンに関する情報がすべて削除されました。
- サンプルの DB2® ストアード・プロシージャが新しい ELAXF* ビルド・プロシージャに置き換わりました。これに伴い、DB2 ストアード・プロシージャに関するすべての情報が削除されました。
- サポートされていないリリースのマイグレーション情報が削除されました。

本書には、「*IBM Rational Developer for System z* バージョン 8.5.1 ホスト構成ガイド」(SC88-5663-06) に記載されていた情報が含まれています。

新しい情報:

- バージョン 8.5.1 でのマイグレーションに関する情報。211 ページの『バージョン 8.5 のマイグレーションに関する注』を参照してください。
- rsed.envvars 内の新しいディレクティブおよび変更されたディレクティブ。43 ページの『rsed.envvars、RSE 構成ファイル』を参照してください。
- CA Endeavor® SCM RAM バッチ・アクションについて拡張されたサポート。100 ページの『CA Endeavor® SCM RAM バッチ・アクション』を参照してください。
- インストール・ユーザー ID の要件。5 ページの『インストール・ユーザー ID』を参照してください。

本書には、「*IBM Rational Developer for System z* バージョン 8.5 ホスト構成ガイド」(SC88-5663-05) に記載されていた情報が含まれています。

新しい情報:

- FEJJCENFG 内の新しいオプションのディレクティブ。38 ページの『FEJJCENFG、JES ジョブ・モニター構成ファイル』を参照してください。
- rsed.envvars 内の新しいオプションのディレクティブ。43 ページの『rsed.envvars、RSE 構成ファイル』を参照してください。
- 新しいオプションの構成ファイル。137 ページの『(オプション) include.conf、C/C++ コンテンツ・アシスト用の強制インクルード』を参照してください。
- 新しいオプションのコンポーネント。139 ページの『(オプション) インクルード・プリプロセッサのサポート』を参照してください。
- 新しいオプションのコンポーネント。140 ページの『(オプション) Enterprise COBOL および PL/I での xUnit サポート』を参照してください。
- 新しいオプションのコンポーネント。152 ページの『(オプション) DB2 および IMS のデバッグのサポート』を参照してください。

- 新しいオペレーター・コマンドおよび拡張されたオペレーター・コマンド。221 ページの『第 11 章 オペレーター・コマンド』を参照してください。
- バージョン 8.5 でのマイグレーション情報。213 ページの『バージョン 8.0.1 からバージョン 8.5 へのマイグレーション』を参照してください。

削除された情報:

- File Manager Integration のサポートが変更されたため、153 ページの『(オプション) File Manager のサポート』のセクションのほとんどの情報が削除されています。

本書には、「*IBM Rational Developer for System z Version 8.0.3 Host Configuration Guide*」(SC23-7658-06)に記載されていた情報が含まれています。

新しい情報:

- rsed.envvars 内の新しいディレクティブ。43 ページの『rsed.envvars、RSE 構成ファイル』を参照してください。
- CA Endeavor® SCM バックグラウンド・アクションのサポートが追加されました。100 ページの『CA Endeavor® SCM RAM バッチ・アクション』を参照してください。
- CA Endeavor® SCM パッケージのサポートが追加されました。95 ページの『CARMA Repository Access Manager (RAM)』を参照してください。
- pushtoclient.properties 内の新しいディレクティブ。129 ページの『(オプション) pushtoclient.properties、ホスト・ベースのクライアント制御』を参照してください。
- File Manager Integration が非推奨になりました。153 ページの『(オプション) File Manager のサポート』を参照してください。
- 新しいオペレーター・コマンドおよび拡張されたオペレーター・コマンド。221 ページの『第 11 章 オペレーター・コマンド』を参照してください。
- 新しい資料、「*IBM Rational Developer for System z メッセージとコード・ガイド*」(SA88-4565)。

本書には、「*IBM Rational Developer for System z バージョン 8.0.1 ホスト構成ガイド*」(SC88-5663-03)に記載されていた情報が含まれています。

新しい情報:

- FEJJCENFG 内の新しいディレクティブ。38 ページの『FEJJCENFG、JES ジョブ・モニター構成ファイル』を参照してください。
- rsed.envvars 内の新しいディレクティブ。43 ページの『rsed.envvars、RSE 構成ファイル』を参照してください。
- CARMA に関する章が再編成され、新しい情報がいくつか追加されました。71 ページの『第 3 章 (オプション) 共通アクセス・リポジトリ・マネージャー (CARMA)』を参照してください。
- 新しい構成ファイル pushtoclient.properties。129 ページの『(オプション) pushtoclient.properties、ホスト・ベースのクライアント制御』を参照してください。

削除された情報:

- 「*IBM Rational Developer for System z* バージョン 7.6.1 ホスト構成ガイド」(SC88-5663-02) に収録されていた内容は、「*IBM Rational Developer for System z* ホスト構成ガイド」(SC88-5663) と「*IBM Rational Developer for System z* ホスト構成リファレンス」(SA88-4226) の 2 冊に分割されました。
- APPC セットアップに関する情報は、「*Using APPC to provide TSO command services*」(SC14-7291) という表題のホワイト・ペーパーに移されました。
- ISPF クライアント・ゲートウェイを使用する CARMA に関する情報は、「*Using ISPF Client Gateway to provide CARMA services*」(SC14-7292) という表題のホワイト・ペーパーに移されました。
- 『(オプション) その他のカスタマイズ・タスク』のセクション『(オプション) ホスト・ベース・プロパティー・グループ』(propertiescfg.properties についての説明)。
- 『(オプション) その他のカスタマイズ・タスク』のセクション『(オプション) ホスト・ベース・プロジェクト』(projectcfg.properties についての説明)。
- 『(オプション) その他のカスタマイズ・タスク』のセクション『(オプション) 編集不可能文字』(uchars.settings についての説明)。
- 『マイグレーション・ガイド』のセクション『バージョン 7.6.1 のマイグレーションに関する注』。

文書内容の説明

このセクションでは、本書に記載する情報を要約します。

計画

この章の情報をを使用して、Developer for System z のインストールとデプロイメントを計画してください。

基本的なカスタマイズ

以下のカスタマイズ・ステップは、基本的な Developer for System z セットアップ用です。

- 20 ページの『カスタマイズのセットアップ』
- 21 ページの『PARMLIB の変更』
- 29 ページの『PROCLIB の変更』
- 37 ページの『セキュリティ定義』
- 38 ページの『FEJCNFG、JES ジョブ・モニター構成ファイル』
- 43 ページの『rsed.envvars、RSE 構成ファイル』
- 67 ページの『ISPF.conf、ISPF の TSO/ISPF クライアント・ゲートウェイ構成ファイル』

(オプション) 共通アクセス・リポジトリ・マネージャー (CARMA)

共通アクセス・リポジトリ・マネージャー (CARMA) は、Repository Access Manager (RAM) 向けのサーバー・プラットフォームです。RAM は、z/OS システム・ベースの Software Configuration Manager (SCM) 用のアプリケーション・プロ

グラミング・インターフェース (API) です。SCM 機能を RAM に内包することにより、サポートされる SCM にクライアントがアクセスするときに、単一の API が使用できるようになります。

Developer for System z は事前に組み込まれた複数の RAM と、ユーザーが独自の RAM を作成するためのソース・コード・サンプルを提供しています。

IBM Rational Developer for System z Interface for CA Endeavor® Software Configuration Manager は、Developer for System z クライアントが CA Endeavor® SCM に直接アクセスできるようにします。

(オプション) SCLM Developer Toolkit

SCLM Developer Toolkit は、SCLM の機能を拡張するために必要なツールをクライアントに提供します。SCLM 自体はホスト・ベースのソース・コード・マネージャーであり、ISPF に組み込まれています。

SCLM Developer Toolkit は、SCLM へのインターフェースとして機能する Eclipse ベースのプラグインを備えています。これにより、従来型コード開発におけるすべての SCLM プロセスへのアクセスを可能にし、メインフレーム上の SCLM と同期したワークステーション上で完全な Java™ および J2EE 開発を行えるようにサポートします。同期アクティビティには、メインフレームからの J2EE コードのビルド、アセンブル、およびデプロイメントが含まれます。

(オプション) Application Deployment Manager (非推奨)

Developer for System z は、さまざまなコンポーネントのための共通のデプロイメント方法として、Application Deployment Manager の特定の機能を使用します。オプションのカスタマイズにより、より多くの Application Deployment Manager のフィーチャーが使用可能になり、以下のサービスを Developer for System z に追加できます。

- IBM CICS エクスプローラーは、CICS リソースを表示および管理するための Eclipse ベースのインフラストラクチャーを提供し、CICS ツール同士のさらに緊密な統合を可能にします。
- CICS リソース定義 (CRD) クライアントおよびサーバーは、以下の機能を提供します。
 - CICS リソース定義エディター
 - CICS リソースを、制限付きの制御されたセキュアな方法でアプリケーション開発者が定義する
 - CICS 管理者がファイル定義内の物理データ・セット名属性を制御できるようにして、無許可または不正な VSAM データ・セットへの CICS 開発アクセスを防止する
 - 各種の CICS 開発援助機能
 - 各種の CICS Web サービス開発援助機能

(オプション) ホスト・ベースのコード分析

Developer for System z クライアントと同様に、Developer for System z ホストは別の製品として提供されている実行中のコード分析ツール、IBM Rational Developer

for System z Host Utilities をサポートします。このコード分析をホスト上で実行する利点は、それを日常のバッチ処理として組み込むことができる点です。

以下のコード分析ツールを、ホスト上で使用することができます。

- コード・レビュー: コード・レビューは、異なる重大度レベルを持つ規則を使用して、ソース・コードをスキャンし、規則違反を報告します。
- コード・カバレッジ: 実行中のプログラムを分析し、実行可能な行の総数に対する実行済みの行についてのレポートを生成します。

(オプション) その他のカスタマイズ・タスク

このセクションは、さまざまなオプションのカスタマイズ・タスクを結合したものです。必要なサービスを構成するには、該当するセクションの説明に従ってください。

Developer for System z 構成ファイルへのカスタマイズ

- pushtoclient.properties、ホスト・ベースのクライアント制御
- ssl.properties、RSE SSL 暗号化
- rsecomm.properties、RSE トレース
- include.conf、C/C++ コンテンツ・アシスト用の強制インクルード

他の製品への Developer for System z 関連のカスタマイズ:

- DB2 ストアード・プロシージャ
- z/OS UNIX サブプロジェクト
- インクルード・プリプロセッサのサポート
- Enterprise COBOL および PL/I での xUnit サポート
- エンタープライズ・サービス・ツール・サポート
- CICS 双方向言語サポート
- 生成コードの診断用 IRZ メッセージ
- 統合デバッガー
- 問題判別ツールのサポート
- DB2 および IMS™ のデバッグのサポート
- File Manager のサポート
- WORKAREA と /tmp のクリーンアップ

インストール検査

製品のカスタマイズの完了後、この章で説明するインストール検査プログラム (IVP) を使用して、主要な製品コンポーネントのセットアップが正常であることを検査できます。

セキュリティ定義

ここでは、サンプルの RACF® コマンドを使用して必須およびオプションのセキュリティ定義について説明します。

マイグレーション・ガイド

ここでは、本製品の以前のリリースと比較したインストールおよび構成上の変更点に重点を置いて説明します。また、このリリースへのマイグレーションに関する一般的なガイドラインも示します。

オペレーター・コマンド

このセクションでは、Developer for System z で使用可能なオペレーター（またはコンソール）コマンドの概要を説明します。

ホスト構成リファレンス

ここでは、「*IBM Rational Developer for System z* ホスト構成リファレンス」(SA88-4226) に記載されている情報を要約します。

IBM Rational Developer for System z ホスト構成ガイド

第 1 章 計画

この章の情報は、Developer for System z のインストールとデプロイメントを計画する場合に、「*IBM Rational Developer for System z 前提条件*」(SC88-4704) の情報と一緒に使用してください。以下の事柄について説明しています。

- 『マイグレーションに関する考慮事項』
- 『計画に関する考慮事項』
- 4 ページの『インストール前の考慮事項』
- 11 ページの『構成前の考慮事項』
- 14 ページの『デプロイメント前の考慮事項』
- 15 ページの『クライアント・チェックリスト』

マイグレーションに関する考慮事項

189 ページの『第 10 章 マイグレーション・ガイド』では、本製品の以前のリリースと比較したインストールおよび構成の変更点を説明します。この情報を使用して、Developer for System z の現行リリースへのマイグレーションを計画してください。

注:

- 以前に IBM Rational Developer for System z、IBM WebSphere Developer for System z、IBM WebSphere Developer for zSeries、または IBM WebSphere Studio Enterprise Developer を使用していた場合は、IBM Rational Developer for System z バージョン 9.1.1 をインストールする前に、関連するカスタマイズ済みのファイルを保存してください。カスタマイズが必要だったファイルの概要については、189 ページの『第 10 章 マイグレーション・ガイド』を参照してください。
- Developer for System z の複数のインスタンスを実行する計画の場合は、「*ホスト構成リファレンス*」(SA88-4226) の『複数のインスタンスの実行』を参照してください。

計画に関する考慮事項

製品の概要

Developer for System z は、ユーザーのパーソナル・コンピューターにインストールされたクライアント、および 1 台以上のホスト・システムにインストールされたサーバーで構成されます。この資料では、z/OS ホスト・システムに関する情報を記載しています。ただし、AIX® や Linux on System z などの他のオペレーティング・システムもサポートされています。

クライアントでは開発者は Eclipse ベースの開発環境を利用できます。この開発環境は、ホストに対するグラフィカル・インターフェースの統一を促進します。また、何よりも、ホストからクライアントへの作業のオフロードを可能にしてホスト上のリソースを節約します。

ホスト部分は、永続的にアクティブな各種のタスクと、臨時に開始されるタスクから構成されます。これらのタスクによって、クライアントは z/OS ホスト・システムのさまざまなコンポーネント (MVS データ・セット、TSO コマンド、z/OS UNIX のファイルとコマンド、ジョブ実行依頼、ジョブ出力など) を処理することができます。

Developer for System z は、ホスト・システム上のサブシステムや他のアプリケーション・ソフトウェア (CICS、IBM File Manager、Software Configuration Manager (SCM) など) と連携することもできます。そのためには、Developer for System z がそのように機能するよう構成されていることと、相互に必要なこれらの製品が使用可能であることが前提となります。

Developer for System z の設計に関する基礎知識については、「ホスト構成リファレンス」(SA88-4226) の『Developer for System z について』を参照してください。

Developer for System z に備わっている機能の詳細については、Developer for System z の Web サイト (<http://www-03.ibm.com/software/products/en/developerforsystemz/>) を参照するか、IBM 担当員にお問い合わせください。

スキルの要件

Developer for System z ホストをインストールするには、SMP/E のスキルが必要です。

Developer for System z を構成するために必要なシステム・プログラミングの権限と専門知識は標準的なレベルを超えているため、他のユーザーからの支援が必要になる場合があります。8 ページの表 3、および 9 ページの表 4に、必須およびオプションのカスタマイズ・タスクに必要な管理者がリストされています。

時間の要件

Developer for System z ホスト・システム・コンポーネントのインストールと構成にかかる時間は、以下のようなさまざまな要因によって決まります。

- z/OS UNIX および TCP/IP の現行の構成
- 前提ソフトウェアと保守の可用性
- Developer for System z ユーザーに対する OMVS セグメントの定義の有無
- クライアントのインストールが完了しているユーザーの有無 (インストール済み環境をテストして発生のある可能性がある問題を報告するため)

経験則から判断すると、Developer for System z ホスト・システムをインストールして構成するプロセスには、完了までに 1 日から 4 日が必要です。これは、経験を積んだシステム・プログラマーが新規インストールを行う場合の時間要件です。問題が発生したり、必要とされるスキルが不足していたりする場合は、セットアップにさらに時間がかかります。

インストール前の考慮事項

SMP/E による製品のインストールの詳細な手順については、「*Program Directory for IBM Rational Developer for System z*」(GI88-4172) を参照してください。

Developer for System z サーバーは、単一システム志向であり、SYSPLEX 対応ではありません。SYSPLEX 内のサーバーを使用している場合、エンド・ユーザーの要求したデータ (データ・セット、ジョブ出力、z/OS UNIX ファイル) が、Developer for System z がインストールされているシステムにあることを確認する必要があります。他のシステムで Developer for System z のクローン作成を行う方法については、14 ページの『デプロイメント前の考慮事項』を参照してください。

単一ホスト・システム上で Developer for System z の複数のインスタンスを実行する場合は、「ホスト構成リファレンス」(SA88-4226) の『複数のインスタンスの実行』を参照してください。

Developer for System z がインストールされているファイル・システム (HFS または zFS) は、SETUID 許可ビットをオン (これはデフォルトです) にしてマウントする必要があります。NOSETUID パラメーターを指定してファイル・システムをマウントすると、ユーザーのセキュリティ環境が Developer for System z によって作成されず、クライアントの接続要求が拒否されます。Java および z/OS UNIX バイナリーをホスティングしているファイル・システムについても同様のことが言えます。

インストール・ユーザー ID

Developer for System z のインストール、またはメンテナンスのインストールに使用するユーザー ID には、最低でも以下の属性が必要です。

- TSO アクセス (通常の領域サイズ)。

注: インストール検査プログラム (IVP) を実行するユーザー ID には、大きな領域サイズが必要です。これは、大量のメモリーを必要とする機能 (Java など) が実行されるからです。領域サイズは、131072 キロバイト (128 メガバイト) 以上に設定してください。

- ユーザー ID とそのデフォルト・グループの両方に対して、セキュリティ・システム (RACF など) に定義された OMVS セグメント
 - HOME フィールドは、ユーザーに READ、WRITE、および EXECUTE アクセス権付きで割り振られたホーム・ディレクトリーを参照していなければなりません。
 - OMVS セグメント内の PROGRAM フィールドは、/bin/sh とするか、または、それ以外の有効な z/OS UNIX シェル (/bin/tcsh など) にしてください。
 - ユーザー ID のデフォルト・グループには GID が必要です。
- UID=0 または FACILITY クラスの BPX.SUPERUSER プロファイルの READ 権限。
- BPX.FILEATTR.APF または BPX.FILEATTR.PROGCTL プロファイルが FACILITY クラスで定義されている場合は、これらのプロファイルの READ アクセス権。
- /tmp ディレクトリー (または TMPDIR 環境変数で参照されるディレクトリー) への READ、WRITE、および EXECUTE アクセス権

必要な製品

Developer for System z には、製品が機能する前にインストールして操作可能にしておく必要がある、前提ソフトウェアがリストされています。また、Developer for System z の特定のフィーチャーをサポートするために相互に必要なソフトウェア

アのリストもあります。該当するフィーチャーを設計どおりに機能させるには、これらの必要なソフトウェアをインストールし、実行時に操作可能になるようにしておく必要があります。

「IBM Rational Developer for System z 前提条件ガイド」(SC88-4704) には、Developer for System z を機能させる前にインストールして操作可能にしておく必要がある、前提ソフトウェアがリストされています。また、Developer for System z の特定のフィーチャーをサポートするために相互に必要なとなるソフトウェアのリストもあります。該当するフィーチャーを設計どおりに機能させるには、これらの必要なソフトウェアをインストールし、実行時に操作可能になるようにしておく必要があります。この資料の最新バージョンは、Developer for System z Web サイト (http://www-01.ibm.com/software/sw-library/en_US/products/Z964267S85716U24/) のライブラリー・ページにあります。

ご使用のサイトのポリシーによっては、そのために少し時間を要する場合もあります。これらの必要な製品が使用可能になるように、事前に計画を立ててください。次に、基本セットアップの主要な必要条件を示します。

- z/OS 1.12 以上
- Java 6.0 以上 (31 または 64 ビット) の最新のサービス・リリース

必要なリソース

Developer for System z を使用するには、表 1 にリストしたシステム・リソースの割り振りが必要です。オプションのサービスには、7 ページの表 2 にリストしたリソースが必要です。これらのリソースを使用できるようにするための計画を立ててください。これは、ご使用のサイトのポリシーによっては、ソフトウェアの取得に一定程度の時間がかかる場合があるためです。

表 1. 必要なリソース

リソース	デフォルト値	情報
LPA データ・セット	FEK.SFEKLPA	24 ページの『LPALSTxx での LPA 定義』
APF 許可データ・セット	FEK.SFEKAUTH	25 ページの『PROGxx での APF 許可』
開始タスク	JMONおよび RSED	29 ページの『PROCLIB の変更』
ホスト限定使用のポート (JMON)	6715	38 ページの『FEJJCNFG、JES ジョブ・モニター構成ファイル』
クライアント/ホスト通信用のポート (RSED)	4035	43 ページの『rsed.envvars、RSE 構成ファイル』
クライアント/ホスト通信用のポート範囲 (RSED)	使用可能な任意のポートを使用	55 ページの『RSE サーバーに使用可能な PORTRANGE の定義』
z/OS UNIX サーバー・セキュリティ定義	RSED 開始タスクについての BPX.SERVER に対する UPDATE 権限	176 ページの『セキュアな z/OS UNIX サーバーとして RSE を定義する』

表 1. 必要なリソース (続き)

リソース	デフォルト値	情報
PassTicket セキュリティー定義	デフォルトなし	177 ページの『RSE の PassTicket サポートの定義』
MVS ビルド・プロシージャ	ELAXF*	29 ページの『PROCLIB の変更』

表 2. オプションのリソース

リソース	デフォルト値	情報
CLPA を伴う IPL	適用外	144 ページの『(オプション) 統合デバッガー』
開始タスク	DBGMR	144 ページの『(オプション) 統合デバッガー』
LINKLIST データ・セット	FEK.SFEKAUTH および FEK.SFEKLOAD	<ul style="list-style-type: none"> 109 ページの『第 4 章 (オプション) SCLM Developer Toolkit』 144 ページの『(オプション) 統合デバッガー』
LPA データ・セット	FEK.SFEKLPA	<ul style="list-style-type: none"> 24 ページの『LPALSTxx での LPA 定義』 71 ページの『第 3 章 (オプション) 共通アクセス・リポジトリ・マネージャ (CARMA)』 144 ページの『(オプション) 統合デバッガー』
セキュリティ・プロファイル	AQE.**	144 ページの『(オプション) 統合デバッガー』
ホスト限定使用のポート範囲	使用可能な任意のポートを使用	<ul style="list-style-type: none"> 71 ページの『第 3 章 (オプション) 共通アクセス・リポジトリ・マネージャ (CARMA)』 125 ページの『第 6 章 (オプション) ホスト・ベースのコード分析』
ホスト限定使用のポート範囲	5336	144 ページの『(オプション) 統合デバッガー』
クライアント/ホスト通信用のポート	<ul style="list-style-type: none"> 5129 (Web サービスの場合) または 5130 (RESTful サービスの場合) 5335 (統合デバッガーの場合) 	<ul style="list-style-type: none"> 117 ページの『第 5 章 (オプション) Application Deployment Manager (非推奨)』 144 ページの『(オプション) 統合デバッガー』

表 2. オプションのリソース (続き)

リソース	デフォルト値	情報
CICS CSD アップデート	複数値	<ul style="list-style-type: none"> 117 ページの『第 5 章 (オプション) Application Deployment Manager (非推奨)』 144 ページの『(オプション) 統合デバッガー』
CICS JCL アップデート	<ul style="list-style-type: none"> FEK.SFEKLOAD FEK.SFEKAUTH 	<ul style="list-style-type: none"> 117 ページの『第 5 章 (オプション) Application Deployment Manager (非推奨)』 142 ページの『(オプション) CICS 双方向言語サポート』 144 ページの『(オプション) 統合デバッガー』

Developer for System z を構成するために必要なシステム・プログラミングの権限と専門知識は標準的なレベルを超えているため、他のユーザーからの支援が必要になる場合があります。表 3、および 9 ページの表 4に、必須およびオプションのカスタマイズ・タスクに必要な管理者がリストされています。

表 3. 必須タスクに必要な管理者

管理者	タスク	情報
システム	すべてのカスタマイズ・タスクには、一般的なシステム・プログラマー・アクションが必要である	適用外

表 3. 必須タスクに必要な管理者 (続き)

管理者	タスク	情報
セキュリティ	<ul style="list-style-type: none"> • Developer for System z ユーザーの OMVS セグメントを定義する • データ・セット・プロファイルを定義する • 開始タスクを定義する • オペレーター・コマンド・セキュリティを定義する • z/OS UNIX サーバー・プロファイルを定義する • アプリケーション・セキュリティを定義する • PassTicket サポートを定義する • プログラム制御データ・セットを定義する • プログラム制御 z/OS UNIX ファイルを定義する 	「ホスト構成リファレンス」(SA88-4226) の『セキュリティに関する考慮事項』
TCP/IP	新しい TCP/IP ポートを定義する	「ホスト構成リファレンス」(SA88-4226) の『TCP/IP に関する考慮事項』
WLM	開始タスクの目標をサーバーとその子プロセスに割り当てる	「ホスト構成リファレンス」(SA88-4226) の『WLM に関する考慮事項』

表 4. オプションのタスクに必要な管理者

管理者	タスク	情報
システム	すべてのカスタマイズ・タスクには、一般的なシステム・プログラマー・アクションが必要である	適用外

表 4. オプションのタスクに必要な管理者 (続き)

管理者	タスク	情報
セキュリティ	<ul style="list-style-type: none"> データ・セット・プロファイルを定義する プログラム制御データ・セットを定義する xxx* ジョブを実行依頼するための許可を定義する CICS トランザクション・セキュリティを定義する SSL の証明書を追加する X.509 クライアント証明書サポートを定義する クライアントへのプッシュ用のグループおよびプロファイルを定義する クライアント機能を変更するためのプロファイルを定義する 開始タスクを定義する z/OS UNIX サーバー・プロファイルを定義する デバッグ用のプロファイルを定義する 	<ul style="list-style-type: none"> 「ホスト構成リファレンス」(SA88-4226) の『セキュリティに関する考慮事項』 「ホスト構成リファレンス」(SA88-4226) の『CICSTS に関する考慮事項』 「ホスト構成リファレンス」(SA88-4226) の『SSL および X.509 認証のセットアップ』
TCP/IP	新しい TCP/IP ポートを定義する	「ホスト構成リファレンス」(SA88-4226) の『TCP/IP ポート』
SCLM	<ul style="list-style-type: none"> Java EE サポート用の SCLM 言語変換プログラムを定義する Java EE サポート用の SCLM タイプを定義する 	109 ページの『第 4 章 (オプション) SCLM Developer Toolkit』
CICS TS	<ul style="list-style-type: none"> CICS 領域 JCL を更新する CICS 領域 CSD を更新する CICS グループを定義する CICS トランザクション名を定義する CICS に対してプログラムを定義する CICS に対してデバッガーを定義する 	<ul style="list-style-type: none"> 117 ページの『第 5 章 (オプション) Application Deployment Manager (非推奨)』 142 ページの『(オプション) CICS 双方向言語サポート』 144 ページの『(オプション) 統合デバッガー』

表 4. オプションのタスクに必要な管理者 (続き)

管理者	タスク	情報
WLM	<ul style="list-style-type: none"> Developer for System z タスクに最終目標を割り当てる 	<ul style="list-style-type: none"> 「ホスト構成リファレンス」 (SA88-4226) の『WLM に関する考慮事項』
LDAP	クライアントへのプッシュ用のグループを定義する	「ホスト構成リファレンス」 (SA88-4226) の『クライアントへのプッシュ機能に関する考慮事項』

構成前の考慮事項

Developer for System z 自体の情報、またそれがシステムと連携する方法、ならびに前提条件および相互必要条件となる製品と連携する方法については、「ホスト構成リファレンス」 (SA88-4226) を参照してください。この情報は、現在のニーズや今後の成長をサポートするようなセットアップを作成する上で助けとなります。

ワークロード管理

従来の z/OS アプリケーションとは異なり、Developer for System z は、ワークロード・マネージャー (WLM) で容易に識別できる一体構造のアプリケーションではありません。Developer for System z は、クライアントがホスト・システムのサービスとデータにアクセスできるようにするために相互に作用する、複数のコンポーネントで構成されています。WLM 構成を計画するには、「ホスト構成リファレンス」 (SA88-4226) の『WLM に関する考慮事項』を参照してください。

注: Developer for System z は、相互に、およびクライアントと通信する複数のタスクから構成されています。これらのタスクでは、さまざまなタイマーを使用してパートナーとの通信が失われたことを検出します。CPU の負荷が高いシステムや、Developer for System z に対するワークロード管理 (WLM) の設定が正しくないシステムでは、タイムアウトの問題が発生する可能性があります (タイムアウト・ウィンドウ内で CPU 時間が不足するため)。

リソース使用量とシステム限度

Developer for System z では、アドレス・スペースや z/OS UNIX のプロセスおよびスレッドなど、不定数のシステム・リソースが使用されます。これらのリソースの可用性は、さまざまなシステム定義によって制限されます。システム構成を計画するために主要なリソースの使用量を見積もるには、「ホスト構成リファレンス」 (SA88-4226) の『チューニングに関する考慮事項』を参照してください。Developer for System z は、ストレージ・リソースの制限を大幅に変更すれば、31 ビットまたは 64 ビット・モードのいずれでも実行できます。

必要な製品の必須の構成

MVS システム・プログラマー、セキュリティー管理者、および TCP/IP 管理者に、必要な製品とソフトウェアがインストールされ、テストされ、機能しているかどうか

かを確認してください。必要であるのに見過ごされる可能性があるカスタマイズ・タスクのいくつかを、以下にリストします。

- すべての Developer for System z ユーザーが、Java ディレクトリーに対する READ アクセス権と EXECUTE アクセス権を持っている必要があります。
- z/OS UNIX サブプロジェクトのリモート (ホスト・ベースの) アクションを実行するには、ホスト・システム上で z/OS UNIX バージョンの REXEC または SSH がアクティブであることが必要です。

ユーザー ID に関する考慮事項

Developer for System z ユーザーのユーザー ID には、以下の属性が最低でも必要です。

- TSO アクセス (通常の領域サイズ)

注: インストール検査プログラム (IVP) を実行するユーザー ID には、大きな領域サイズが必要です。これは、大量のメモリを必要とする機能 (Java など) が実行されるからです。領域サイズは、131072 キロバイト (128 メガバイト) 以上に設定してください。

- ユーザー ID とそのデフォルト・グループの両方に対して、セキュリティ・システム (RACF など) に定義された OMVS セグメント
 - HOME フィールドは、ユーザーに (READ、WRITE、および EXECUTE アクセス権付で) 割り振られたホーム・ディレクトリーを参照する必要があります。
 - OMVS セグメント内の PROGRAM フィールドは、/bin/sh とするか、または、それ以外の有効な z/OS UNIX シェル (/bin/tcsh など) にしてください。
 - ASSIZEMAX フィールドは設定せず、システム・デフォルトが使用されるようにしてください。
 - ユーザー ID は UID 0 を必要としません。

例 (コマンド **LISTUSER userid NORACF OMVS**):

```
USER=userid
```

```
OMVS INFORMATION
```

```
-----  
UID= 0000003200  
HOME= /u/userid  
PROGRAM= /bin/sh  
CPUTIMEMAX= NONE  
ASSIZEMAX= NONE  
FILEPROCMA= NONE  
PROCUSERMA= NONE  
THREADSMA= NONE  
MMAPAREAMA= NONE
```

- ユーザー ID のデフォルト・グループには GID が必要です。

例 (コマンド **LISTGRP group NORACF OMVS**):

```
GROUP group
```

```
OMVS INFORMATION
```

```
-----  
GID= 0000003243
```

- Developer for System z インストールおよび構成ディレクトリーとファイル (デフォルトでは /usr/lpp/rdz/*、/etc/rdz/*、および /var/rdz/*) への READ および EXECUTE アクセス権
- Developer for System z WORKAREA ディレクトリー (デフォルトでは /var/rdz/WORKAREA) およびユーザー・ログ・ディレクトリー (デフォルトでは /var/rdz/logs) への READ、WRITE、および EXECUTE アクセス権
- Developer for System z インストール・データ・セット (デフォルトでは FEK.SFEK*) への READ アクセス権
- /tmp ディレクトリーまたは TMPDIR 環境変数で参照されるディレクトリーへの READ、WRITE、および EXECUTE アクセス権

サーバーに関する考慮事項

Developer for System z は永続的にアクティブないくつかのサーバーから構成され、これらのサーバーは開始タスクまたはユーザー・ジョブとして機能できます。これらのサーバーは、要求されたサービスをそれら自体が提供するか、他のサーバー (z/OS UNIX スレッドまたはユーザー・ジョブなど) を始動してサービスを提供します。始動順序は特にありません。唯一の要件は、最初のユーザーが接続を試みるまでにサーバーが稼働していることです。Developer for System z のサーバーとサービスによって使用されるセキュリティー・メカニズムは、それらが存在するデータ・セットとファイル・システムが保護されていることを前提とします。つまり、信頼されたシステム管理者のみがプログラム・ライブラリーと構成ファイルを更新できる状態でなければなりません。

- デバッグ・マネージャー (DBGMR) は、デバッグ関連のサービスを提供します。
- JES ジョブ・モニター (JMON) は、JES に関連したすべてのサービスを提供します。
- リモート・システム・エクスプローラー (RSE) は、クライアントをホスト・システムに接続したり、特定のサービス用に他のサーバーを始動するなどの、コア・サービスを提供します。RSE は、次の 2 つの論理エンティティーから構成されます。
 - RSE デーモン (RSED)。これは接続セットアップを管理し、単一サーバー・モードでの実行を担当します。
 - RSE サーバー。これは個々のクライアント要求を処理します。

「ホスト構成リファレンス」(SA88-4226) の『TCP/IP ポート』に説明されているように、特定のホスト・システム・サービスとそれらのポートが、クライアントから接続できるようになっていなければならず、また、ホスト・システムを保護しているファイアウォールに定義されていなければなりません。これ以外に Developer for System z が使用するポートは、すべて、ホスト専用トラフィックを持ちます。以下に、基本的な Developer for System z セットアップでの外部通信に必要なポートを示します。

- クライアント/ホスト通信セットアップ用 (TCP を使用) の RSE デーモン、デフォルト・ポート 4035。
- クライアント/ホスト通信用 (TCP を使用) の RSE サーバー。デフォルトでは、使用可能な任意のポートが使用されますが、使用可能なポートを指定の範囲に制限することができます。

構成方法

Developer for System z は、製品のホスト・システム側を構成するための代替方法を提供します。以下はその方法です。

- ホスト構成ユーティリティーの使用。この ISPF パネルのアプリケーションでは、必要なカスタマイズ・ステップおよび選ばれたオプションのカスタマイズ・ステップを順に実行していきます。詳しくは、「ホスト構成ユーティリティー・ガイド」(SA88-4197) を参照してください。
- 「ホスト構成クイック・スタート・ガイド」を使用する方法。この資料は、必須のカスタマイズ・ステップが順に説明されています。このガイドの適用範囲は、基本セットアップに限定されています。
- 「ホスト構成ガイド」を使用する方法。この資料は、必須のカスタマイズ・ステップとすべてのオプションのカスタマイズ・ステップが順に説明されています。このガイドには、構成可能なすべてのオプションが、デフォルト以外のシナリオも含めて掲載されています。

デプロイメント前の考慮事項

Developer for System z は、インストール済み環境のクローンを別のシステムに作成することをサポートしているため、システムごとに SMP/E によるインストールを行う必要がありません。

以下のデータ・セット、ディレクトリー、およびファイルは、他のシステムへのデプロイメントに必須です。ファイルを別のロケーションへコピーしてある場合は、下記のリスト内の相当するファイルをそのファイルに置き換える必要があります。

注: 以下のリストでは、前提条件および相互必要条件となるソフトウェアのデプロイメントの必要性については考慮されていません。

Developer for System z

- FEK.SFEKAUTH(*)
- FEK.SFEKLOAD(*)
- FEK.SFEKLPA(*)
- FEK.SFEKPROC(*)
- FEK.#CUST.PARMLIB(*)
- FEK.#CUST.PROCLIB(*)
- /usr/lpp/rdz/*
- /etc/rdz/*
- /var/rdz/* (ディレクトリー構造のみ)
- オプションのパーツ:
 - FEK.SFEKLMOD(*)
 - FEK.#CUST.CNTL(*)
 - FEK.#CUST.JCL でのカスタマイズ・ジョブで生成された定義、データ・セット、ファイル、およびディレクトリー

Developer for System z Host Utilities

- AKG.SAKGPROC(*)
- /usr/lpp/rdzutil/*

注:

- FEK および /usr/lpp/rdz は、Developer for System z のインストール時に使用された高位修飾子およびパスです。FEK.#CUST、/etc/rdz、および /var/rdz は、製品のカスタマイズ時に使用されたデフォルトのロケーションです (詳細については、20 ページの『カスタマイズのセットアップ』を参照してください)。
- AKG および /usr/lpp/rdzutil は、Developer for System z Host Utilities のインストール時に使用された高位修飾子およびパスです。
- Developer for System z では Distributed Dynamic VIPA をサポートしています。これにより、SYSPLEX 内の異なるシステム上にある同じサーバーを単一サーバーとしてクライアントに提示できます。詳しくは、「ホスト構成リファレンス」(SA88-4226) の『Distributed Dynamic VIPA』を参照してください。
- 製品の z/OS UNIX の部分をデプロイしやすくするために、Developer for System z を専用ファイル・システム (HFS または zFS) にインストールしてください。専用ファイル・システムを使用できない場合は、z/OS UNIX の tar コマンドなどのアーカイブ・ツールを使用して z/OS UNIX ディレクトリーをシステム間で転送してください。この方式は、Developer for System z のファイルとディレクトリーの属性 (プログラム制御など) を保存するためのものです。

Developer for System z インストール・ディレクトリーをアーカイブおよび復元するための以下のサンプル・コマンドについて詳しくは、「UNIX System Services コマンド解説書」(SA88-8641) を参照してください。

- アーカイブ: cd /SYS1/usr/lpp/rdz; tar -cSf /u/userid/rdz.tar
- 復元: cd /SYS2/usr/lpp/rdz; tar -xSf /u/userid/rdz.tar

クライアント・チェックリスト

Developer for System z クライアントのユーザーは、クライアントを正しく機能させるために、特定のホスト・システム・カスタマイズの結果 (TCP/IP ポート番号など) を把握しておく必要があります。これらのチェックリストを使用して、必要な情報を収集してください。

表 5 のチェックリストは、必須のカスタマイズ・ステップに必要な結果を示しています。16 ページの表 6 は、オプションのカスタマイズ・ステップに必要な結果を示しています。

表 5. クライアント・チェックリスト: 必須部分

カスタマイズ	値
RSE デーモン TCP/IP ポート番号。デフォルトは 4035 です。	
31 ページの『RSED、RSE デーモン開始タスク』を参照してください。	

表 6. クライアント・チェックリスト: オプション部分

カスタマイズ	値
<p>ELAXF* プロシージャーのロケーション (システム・プロシージャー・ライブラリーに入っていない場合)。デフォルトは FEK.#CUST.PROCLIB です。</p> <p>35 ページの『ELAXF* リモート・ビルド・プロシージャー』で JCLLIB に関する注を参照してください。</p>	
<p>ELAXF* プロシージャーのプロシージャー名またはステップ名 (変更された場合)。</p> <p>35 ページの『ELAXF* リモート・ビルド・プロシージャー』で、それらの変更に関する注を参照してください。</p>	
<p>AKGCR プロシージャーのロケーション (システム・プロシージャー・ライブラリーに入っていない場合)。デフォルトは AKG.#CUST.PROCLIB です。</p> <p>125 ページの『コード・レビュー』で、JCLLIB に関する注を参照してください。</p>	
<p>AKGCC プロシージャーのロケーション (システム・プロシージャー・ライブラリーに入っていない場合)。デフォルトは AKG.#CUST.PROCLIB です。</p> <p>126 ページの『コード・カバレッジ』で、JCLLIB に関する注を参照してください。</p>	
<p>FEKRNPLI インクルード・プリプロセッサ exec ステートメントのロケーション。デフォルトは FEK.#CUST.CNTL です。</p> <p>139 ページの『(オプション) インクルード・プリプロセッサのサポート』を参照してください。</p>	
<p>デバッガー・ロード・モジュールのロケーション (LINKLIST に入っていない場合)。デフォルトは FEK.SFEKAUTH です。 144 ページの『(オプション) 統合デバッガー』を参照してください。</p>	
<p>単体テスト・ロード・モジュールのロケーション (rsed.envvars の LINKLIST または STEPLIB に入っていない場合)。デフォルトは FEK.SFEKLOAD です。</p> <p>140 ページの『(オプション) Enterprise COBOL および PL/I での xUnit サポート』を参照してください。</p>	
<p>AZUZUNIT プロシージャーのロケーション (システム・プロシージャー・ライブラリーに入っていない場合)。デフォルトは FEK.#CUST.PROCLIB です。</p> <p>140 ページの『(オプション) Enterprise COBOL および PL/I での xUnit サポート』で、JCLLIB に関する注を参照してください。</p>	
<p>単体テスト出力のフォーマット設定に使用されるサンプル XML ファイル *.xsd および *.xsl のロケーション。デフォルトは /usr/lpp/rdz/samples/zunit/xsd および /usr/lpp/rdz/samples/zunit/xsl です。</p> <p>140 ページの『(オプション) Enterprise COBOL および PL/I での xUnit サポート』を参照してください。</p>	

表 6. クライアント・チェックリスト: オプション部分 (続き)

カスタマイズ	値
<p>(相互前提条件) Host Connect Emulator の TN3270 ポート番号。デフォルトは 23 です。</p> <p>「ホスト構成リファレンス」(SA88-4226) の『TCP/IP ポート』を参照してください。</p>	
<p>(相互前提条件) REXEC ポート番号または SSH ポート番号。デフォルトはそれぞれ 512 および 22 です。</p> <p>138 ページの『(オプション) z/OS UNIX サブプロジェクト』を参照してください。</p>	
<p>(相互前提条件) Debug Tool サーバーのポート番号 (デフォルトなし)。</p> <p>152 ページの『(オプション) DB2 および IMS のデバッグのサポート』を参照してください。</p>	
<p>Application Deployment Manager ポート番号。デフォルトは 5129 (Web サービスの場合) または 5130 (REST サービスの場合) です。</p> <p>「ホスト構成リファレンス」(SA88-4226) の『TCP/IP ポート』を参照してください。</p>	
<p>CARMA RAM サンプルの SFEKSAMP サンプル・ライブラリーのロケーション。デフォルトは FEK.SFEKSAMP です。</p> <p>「CARMA Developer's Guide」(SC23-7660) を参照してください。</p>	
<p>CARMA SCLM RAM データ・セット割り振りの CRA#ASLM JCL のロケーション。デフォルトは FEK.#CUST.JCL です。</p> <p>97 ページの『SCLM RAM』で CRA#ASLM に関する注を参照してください。</p>	

第 2 章 基本的なカスタマイズ

以下のカスタマイズ・ステップは、基本的な Developer for System z セットアップ用です。オプション・コンポーネントのカスタマイズ要件については、それらのコンポーネントに関する章を参照してください。

要件およびチェックリスト

このカスタマイズ・タスクを完了するには、セキュリティー管理者および TCP/IP 管理者の支援が必要になります。このタスクには、以下のリソースと特殊なカスタマイズ・タスクが必要です。

- LPA データ・セット
- APF 許可データ・セット
- 各種の PARMLIB 更新
- 各種のセキュリティー・ソフトウェア更新
- 内部およびクライアント/ホスト通信用の各種 TCP/IP ポート
- オプションの SVC をアクティブ化するための IPL

インストールを検証し、ご使用のサイトで Developer for System z の使用を開始するためには、以下のタスクを実行してください。特に断りがない限り、すべてのタスクは必須です。

1. カスタマイズ可能なサンプルのコピーを作成し、Developer for System z 用の作業環境を作成します。詳細については、20 ページの『カスタマイズのセットアップ』を参照してください。
2. z/OS UNIX システム限度を更新し、開始タスクを開始します。さらに、APF 許可データ・セットと LINKLIST データ・セットを定義し、オプションで SVC および LPA データ・セットを定義します。詳細については、21 ページの『PARMLIB の変更』を参照してください。
3. 開始タスク・プロシーチャーを作成し、プロシーチャーをコンパイルしてリンクします。詳細については、29 ページの『PROCLIB の変更』を参照してください。
4. セキュリティー定義を更新します。詳細については、37 ページの『セキュリティー定義』を参照してください。スレッド・セキュリティーを設定するには、PassTicket の使用方法を把握する必要があります。「ホスト構成リファレンス」(SA88-4226) の『PassTickets の使用』を参照してください。
5. Developer for System z 構成ファイルをカスタマイズします。詳細については、以下を参照してください。
 - 38 ページの『FEJCNFG、JES ジョブ・モニター構成ファイル』
 - 43 ページの『rsed.envvars、RSE 構成ファイル』
 - 67 ページの『ISPF.conf、ISPF の TSO/ISPF クライアント・ゲートウェイ構成ファイル』

カスタマイズのセットアップ

Developer for System z には、いくつかのサンプル構成ファイルとサンプル JCL が含まれています。カスタマイズした内容がメンテナンスの適用時に上書きされないように、それらのメンバーと z/OS UNIX ファイルをすべて別のロケーションにコピーし、そのコピーをカスタマイズしてください。

Developer for System z の一部の機能は、z/OS UNIX 内に特定のディレクトリーが存在することを必要とします。それらのディレクトリーを製品のカスタマイズ時に作成する必要があります。インストールの労力を軽減するために、コピーと必要なディレクトリーを作成するサンプル・ジョブ、FEKSETUP が提供されています。

注: 「IBM Rational Developer for System z ホスト構成ユーティリティー・ガイド」(SA88-4197) に、ホスト構成ユーティリティーを使用する場合のホスト・システム構成について記載されています。FEKSETUP ジョブとこのユーティリティーは同じタスクをいくつか実行しますが、それらのタスクが既に実行されたかどうかを確認する方法はありません。したがって、既に行われた変更を元に戻すことが可能です。このため、単一のインストールで両方の方式を使用しないでください。

構成ファイルおよび構成 JCL のカスタマイズ可能なコピーを作成し、必要な z/OS UNIX ディレクトリーを作成するには、FEK.SFEKSAMP データ・セット内のサンプル・メンバー FEKSETUP をカスタマイズして実行依頼します。必要なカスタマイズ・ステップは、このメンバー内に記述されています。

このジョブは、以下のタスクを実行します。

- FEK.#CUST.PARMLIB を作成し、これにサンプル構成ファイルを取り込みます。
- FEK.#CUST.PROCLIB を作成し、これにサンプル SYS1.PROCLIB メンバーを取り込みます。
- FEK.#CUST.JCL を作成し、これにサンプル構成 JCL を取り込みます。
- FEK.#CUST.CNTL を作成し、これにサンプルのサーバー始動スクリプトを取り込みます。
- FEK.#CUST.ASM を作成し、これにサンプル・アセンブラー・ソース・コードを取り込みます。
- FEK.#CUST.COBOL を作成し、これにサンプル COBOL ソース・コードを取り込みます。
- FEK.#CUST.SQL を作成し、これにサンプル SQL コマンド・ファイルを取り込みます。
- /etc/rdz/* を作成し、これにサンプルの構成ファイルを取り込みます。
- /var/rdz/* を Developer for System z の各種機能の作業ディレクトリーとして作成し、これにサンプル・ファイルを取り込みます。

注:

- この資料の構成ステップでは、特に断りがない限り、FEKSETUP ジョブによって作成されたメンバーおよびファイル・ロケーションを使用します。更新不可のオリジナルのサンプルは、FEK.SFEKSAMP および /usr/lpp/rdz/samples/ に入っています。

- どのサンプル・メンバーがどのデータ・セットにコピーされるか、どのディレクトリーが作成されるか、それらの許可ビット・マスク、および各種サンプル・ファイルがどこにコピーされるかについての詳細は、FEK.SFEKSAMP(FEKSETUP) 内のコメントを参照してください。
- FEK.SFEKSAMP(FEKSETUP) 内のコメントには、既存のセットアップをマイグレーションする際に役立つように、Developer for System z の異なるバージョン間の変更点についても記載されています。
- すべての Developer for System z z/OS UNIX ファイルを同じファイル・システム (HFS または zFS) に保持したいが、構成ファイルは /etc/rdz に置いておきたいという場合は、シンボリック・リンクを使用して、この問題を解決できます。以下のサンプル z/OS UNIX コマンドは、既存のファイル・システム (/usr/lpp/rdz/cust) 内に新規ディレクトリーを作成し、それへのシンボリック・リンク (/etc/rdz) を定義します。

```
mkdir /usr/lpp/rdz/cust
ln -s /usr/lpp/rdz/cust /etc/rdz
```

PARMLIB の変更

このセクションでは、以下の PARMLIB の変更点について説明します。

- 『BPXPRMxx での z/OS UNIX 限度の設定』
- 23 ページの『COMMNDxx への開始タスクの追加』
- 23 ページの『IEASVCxx での SVC 定義』
- 24 ページの『LPALSTxx での LPA 定義』
- 25 ページの『PROGxx での APF 許可』
- 25 ページの『PROGxx での LINKLIST 定義』
- 27 ページの『必要な LINKLIST 定義と LPA 定義』
- 29 ページの『他の製品用の LINKLIST 定義』

次のセクションにリストされている PARMLIB 定義について詳しくは、「MVS 初期設定およびチューニング解説書」(SA88-8564) を参照してください。サンプルのコンソール・コマンドについて詳しくは、「MVS システム・コマンド」(SA88-8593) を参照してください。

BPXPRMxx での z/OS UNIX 限度の設定

リモート・システム・エクスプローラー (RSE) は、クライアントをホスト・システムに接続するなどのコア・サービスを提供する、z/OS UNIX ベースのプロセスです。したがって、同時にアクティブになる Developer for System z ユーザーの数およびそれらのユーザーの平均的なワークロードに基づいて、BPXPRMxx で z/OS UNIX システム限度に適切な値を設定することが重要です。IPL 時にどの BPXPRMxx parmlib メンバーが使用されるかを指定するには、IEASYSxx parmlib メンバーで OMVS=xx を定義してください。

BPXPRMxx で定義されるさまざまな限度と、それらが Developer for System z に与える影響の詳細については、「ホスト構成リファレンス」(SA88-4226) の『チューニングに関する考慮事項』を参照してください。

MAXASSIZE は、最大アドレス・スペース (プロセス) 領域サイズを指定します。SYS1.PARMLIB(BPXPRMxx) で MAXASSIZE を 2G に設定してください。これは、許容される最大値です。これはシステム全体の限度であるため、すべての z/OS UNIX アドレス・スペースに対してアクティブとなります。これを望まない場合は、セキユーリティー・ソフトウェアで Developer for System z だけに限度を設定することができます。詳細については、174 ページの『Developer for System z 開始タスクの定義』を参照してください。

MAXTHREADS は、単一プロセスでのアクティブなスレッドの最大数を指定します。SYS1.PARMLIB(BPXPRMxx) で MAXTHREADS を 1500 以上に設定してください。これはシステム全体の限度であるため、すべての z/OS UNIX アドレス・スペースに対してアクティブとなります。これを望まない場合は、セキユーリティー・ソフトウェアで Developer for System z だけに限度を設定することができます。詳細については、174 ページの『Developer for System z 開始タスクの定義』を参照してください。

MAXTHREADTASKS は、単一プロセスでのアクティブな MVS タスクの最大数を指定します。SYS1.PARMLIB(BPXPRMxx) で MAXTHREADTASKS を 1500 以上に設定してください。これはシステム全体の限度であるため、すべての z/OS UNIX アドレス・スペースに対してアクティブとなります。これを望まない場合は、セキユーリティー・ソフトウェアで Developer for System z だけに限度を設定することができます。詳細については、174 ページの『Developer for System z 開始タスクの定義』を参照してください。

MAXPROCUSER は、単一の z/OS UNIX ユーザー ID が同時にアクティブにしておくことができるプロセスの最大数を指定します。SYS1.PARMLIB(BPXPRMxx) で MAXPROCUSER を 50 以上に設定してください。この設定は、Developer for System z を使用する各クライアントに対してアクティブにする必要があるため、システム全体の限度となります。

これらの値は、以下のコンソール・コマンドで検査し、動的に (次の IPL まで) 設定できます。

- DISPLAY OMVS,0
- SETOMVS MAXASSIZE=2G
- SETOMVS MAXTHREADS=1500
- SETOMVS MAXTHREADTASKS=1500
- SETOMVS MAXPROCUSER=50

注:

- アドレス・スペース・サイズを設定または制限できる、その他のロケーションについて詳しくは、「ホスト構成リファレンス」(SA88-4226) の『アドレス・スペース・サイズ』を参照してください。
- ここで提案した MAXPROCUSER 値は、固有の z/OS UNIX ユーザー ID (UID) を持つユーザーに基づいています。ユーザーが同じ UID を共用する場合は、この値を大きくしてください。
- その他の BPXPRMxx 値 (MAXPROCSYS や MAXUIDS などの値) を、同時にアクティブになることが予想される Developer for System z ユーザー数を処理するのに十分

な値にしてください。詳しくは、「ホスト構成リファレンス」(SA88-4226)の『チューニングに関する考慮事項』を参照してください。

- Developer for System z を SMP/E でインストールするとき、別個のファイル・システム (HFS の zFS) にコードを配置し、そのファイル・システムがシステム IPL 時にマウントされるように BPXPRMxx を更新することが勧められていました。この更新を依然として必要としている場合のために、マウント・コマンドのサンプルを以下に再掲載します。

```
MOUNT FILESYSTEM('#dsn')
  MOUNTPPOINT('-PathPrefix-usr/lpp/rdz')
  MODE(RDWR) /* can be MODE(READ) */
  TYPE(ZFS) PARM('AGGRGROW') /* zFS, with extents */
/* TYPE(HFS) */ /* HFS, auto. extent */
```

- Developer for System z Host Utilities を SMP/E でインストールするとき、個別のファイル・システム (HFS の zFS) にコードを配置し、そのファイル・システムがシステム IPL 時にマウントされるように BPXPRMxx を更新することが推奨されてきました。この更新を依然として必要としている場合のために、マウント・コマンドのサンプルを以下に再掲載します。

```
MOUNT FILESYSTEM('#dsn')
  MOUNTPPOINT('-PathPrefix-usr/lpp/rdzutil')
  MODE(RDWR) /* can be MODE(READ) */
  TYPE(ZFS) PARM('AGGRGROW') /* zFS, with extents */
/* TYPE(HFS) */ /* HFS, auto. extent */
```

COMMNDxx への開始タスクの追加

Developer for System z RSED および JMON サーバーの始動コマンドを SYS1.PARMLIB(COMMNDxx) に追加し、これらのサーバーが次のシステム IPL で自動的に始動されるようにします。IPL 時にどの COMMNDxx parmlib メンバーが使用されるかを指定するには、IEASYSxx parmlib メンバーで CMD=xx を定義してください。

オプションの統合デバッガーが機能するには、システム上で Developer for System z DBGMR サーバーがアクティブ状態である必要があります。

サーバーを定義および構成した後、これらのサーバーを以下のコンソール・コマンドで動的に (次の IPL まで) 始動できます。

- S RSED
- S JMON
- S DBGMR

注: 他のすべてのサーバーより前に、またはそれらと一緒に、サーバーには特定の始動順序はありません。唯一の要件は、最初のユーザーが接続を試みるまでにサーバーが稼働していることです。

IEASVCxx での SVC 定義

オプションの統合デバッガーは、システムに Developer for System z 監視プログラム呼び出し (SVC) が定義されていることを必要とします。

インストール先定義 SVC は SYS1.PARMLIB(IEASVCxx) に定義され、IPL がアクティブ化されている必要があります。関連するロード・モジュールが IPL 時に LPA

にロードされる必要があります。 IPL 時にどの IEASVCxx parmlib メンバーが使用されるかを指定するには、IEASYSxx parmlib メンバーで SVC=xx を定義してください。

IEASVCxx で以下を指定して Developer for System z SVC を定義します。

```
SVCPARM 251,REPLACE,TYPE(3),EPNAME(AQESVC03) /* RDz debug */
```

SVC 番号 251 がデフォルトですが、z/OS で決められている 200 から 255 までの範囲内であれば、任意の値を使用できます。統合デバッガーは、使用される SVC 番号を検出します。

注: バージョン 9.1.1 より前のバージョンの Developer for System z では、現在使用されていない別の SVC、AQESVC01 を使用していました。これは、ご使用のホスト・システムにバージョン 9.1.1 以上の Developer for System z しかない場合には、削除できます。

LPALSTxx での LPA 定義

Developer for System z では、RSE デーモンでデータ・セットを操作するために、FEK.SFEKLPA ロード・ライブラリー内のモジュールがリンク・パック域 (LPA) 内に存在する必要があります。

オプションの共通アクセス・リポジトリ・マネージャー (CARMA) サービスは、CARMA サーバーに対するさまざまなサーバー始動方式をサポートしています。CRASTART 始動方式では、FEK.SFEKLPA ロード・ライブラリー内のモジュールが LPA 内にいることが必要です。

オプションの統合デバッガーは、FEK.SFEKLPA ロード・ライブラリー内のロード・モジュールが、IPL 時に LPA 内にいることを必要とします。

LPA データ・セットは、SYS1.PARMLIB(LPALSTxx) で定義されます。IPL 時にどの LPALSTxx parmlib メンバーが使用されるかを指定するには、IEASYSxx parmlib メンバーで LPA=xx を定義してください。

以下のコンソール・コマンドで、LPA 定義を動的に (次回の IPL まで) 設定できます。

- SETPROG LPA,ADD,DSN=FEK.SFEKLPA

注:

- LPALSTxx にリストされているデータ・セットは、マスター・カタログか、LPALSTxx メンバーで識別されるユーザー・カタログにカタログされる必要があります。
- 新規データ・セットを LPALSTxx に追加するには、CLPA (LPA の作成) を指定した IPL がアクティブ化されている必要があります。
- LPA にロードされるすべてのライブラリーは、自動的に、APF 許可が与えられていて、プログラム制御されるものとみなされます。これらのライブラリー用に、適切なセキュリティー管理を適切な場所に配置します。
- LPA 配置用に設計されたライブラリーを LPA に配置しないことを選択し、代わりに LINKLIST または STEPLIB を使用する場合、APF 許可およびプログラム制御状況を定義してください。

PROGxx での APF 許可

JES ジョブ・モニターで JES スプール・ファイルにアクセスするには、FEK.SFEKAUTH ロード・ライブラリー内のモジュールおよび CEE.SCEERUN* Language Environment® (LE) ランタイム・ライブラリーに APF 許可が必要です。

オプションのデバッグ・マネージャーを機能させるためには、FEK.SFEKAUTH ロード・ライブラリー内のモジュールに APF 許可が必要です。

オプションの SCLM Developer Toolkit サービスが機能するためには、REXX ランタイム・ライブラリー (REXX.*.SEAGLPA) に APF 許可が必要です。

ISPF で TSO/ISPF クライアント・ゲートウェイを作成するには、SYS1.LINKLIB 内の ISP* モジュールが APF 許可済みである必要があります。TSO/ISPF クライアント・ゲートウェイは、Developer for System z の TSO コマンド・サービスおよび SCLM Developer Toolkit によって使用されます。

APF 許可は、デフォルトでは SYS1.PARMLIB(PROGxx) で定義されます。IPL 時にどの PROGxx parmlib メンバーが使用されるかを指定するには、IEASYSxx parmlib メンバーで PROG=xx を定義してください。

APF 許可は、以下のコンソール・コマンドで動的に (次の IPL まで) 設定できます。ここで、volser はデータ・セットが存在するボリューム (SMS の管理対象でない場合) です。

- SETPROG APF,ADD,DSN=FEK.SFEKAUTH,SMS
- SETPROG APF,ADD,DSN=CEE.SCEERUN,VOL=volser
- SETPROG APF,ADD,DSN=CEE.SCEERUN2,VOL=volser
- SETPROG APF,ADD,DSN=REXX.V1R4M0.SEAGLPA,VOL=volser
- SETPROG APF,ADD,DSN=SYS1.LINKLIB,VOL=volser

注:

- REXX 製品パッケージに代替ライブラリーを使用している場合、デフォルトの REXX ランタイム・ライブラリー名は REXX.*.SEAGALT であり、前のサンプルで使用した REXX.*.SEAGLPA ではありません。
- REXX.*.SEAGLPA などの LPA ライブラリーは、LPA 内に置かれている場合は自動的に APF 許可があり、したがって明示的な定義は必要ありません。
- また、いくつかの相互前提条件となる製品 (IBM File Manager など) も APF 許可を必要とします。詳細については、関連製品のカスタマイズ・ガイドを参照してください。

PROGxx での LINKLIST 定義

Developer for System z の LINKLIST 定義は、以下の 3 つのカテゴリに分けることができます。

- Developer for System z 機能に必要な Developer for System z ロード・ライブラリー。これらの定義については、このセクションで説明します。
- Developer for System z 機能に必要な必須ロード・ライブラリー。これらの定義については、27 ページの『必要な LINKLIST 定義と LPA 定義』で説明します。

- 他の製品が必要とする Developer for System z ロード・ライブラリー。これらの定義については、29 ページの『他の製品用の LINKLIST 定義』で説明します。

表 7. ロード・モジュールと機能のマッチング

ロード・ライブラリー	ロード・モジュール	使用法	STEPLIB
FEK.SFEKAUTH	AQE* および CEE*	144 ページの『(オプション) 統合デバッガー』	ELAXFGO プロシージャー、または CICS
	FEJJ*	29 ページの『PROCLIB の変更』	開始タスク・プロシージャー
FEK.SFEKLMOD	IRZ* および IIRZ*	143 ページの『(オプション) 生成コードの診断用 IRZ メッセージ』	CICS、IMS、または MVS のパッチ
FEK.SFEKLOAD	AND*	117 ページの『第 5 章 (オプション) Application Deployment Manager (非推奨)』	CICS
	AZU* および IAZU*	140 ページの『(オプション) Enterprise COBOL および PL/I での xUnit サポート』	rsed.envvars または MVS パッチ
	BWB*	109 ページの『第 4 章 (オプション) SCLM Developer Toolkit』	rsed.envvars
	CRA*	71 ページの『第 3 章 (オプション) 共通アクセス・リポジトリ・マネージャー (CARMA)』	CRASUB* または crastart*.conf
	ELAX*	35 ページの『ELAXF* リモート・ビルド・プロシージャー』 (エラーのフィードバックおよびインクルード・プリプロセッサ)	ELAXF* プロシージャー
	FEJB*	142 ページの『(オプション) CICS 双方向言語サポート』	CICS
FEK.SFEKLPA	CRA*	71 ページの『第 3 章 (オプション) 共通アクセス・リポジトリ・マネージャー (CARMA)』	CRASRV.properties
	AQE*	144 ページの『(オプション) 統合デバッガー』	適用不可 (LPA は必須)

リストされている Developer for System z サービスが機能するためには、STEPLIB または LINKLIST を通じて、表 7 に示されているサービスに関連するすべてのモジュールを使用可能にする必要があります。SFEKLMOD ライブラリーは、Developer for System z 自体によって使用されるのではなく、Developer for System z によって生成

されたコードによって使用されることに注意してください。STEPLIB を使用する
場合、どこで STEPLIB (または CICS の場合は DFHRPL) を定義する必要があるの
かについては、26 ページの表 7 の STEPLIB の欄を参照してください。ただし、以
下のことに注意する必要があります。

- STEPLIB を z/OS UNIX で使用すると、パフォーマンスに悪い影響が出ます。
- 1 つの STEPLIB ライブラリーに APF 許可がある場合、すべての STEPLIB ラ
イブラリーに APF 許可があることが必要です。ライブラリーは、STEPLIB 内で
許可のないライブラリーと混用した場合、APF 許可を失います。
- JCL で STEPLIB DD に追加されたライブラリーは、その JCL によって開始さ
れた z/OS UNIX プロセスに伝搬されません。

使用しているサイトが IBM の推奨に従っている場合、LINKLIST データ・セット
は、SYS1.PARMLIB(PROGxx) の中で定義されています。IPL 時にどの PROGxx parmlib
メンバーが使用されるかを指定するには、IEASYsxx parmlib メンバーで PROG=xx を
定義してください。

必要な定義は以下のようになります。ここで、listname はアクティブにされる
LINKLIST セットの名前で、volser はマスター・カタログにカタログされていない
データ・セットが存在しているボリュームです。

- LNKLIST ADD NAME(listname) DSN=FEK.SFEKAUTH VOLUME(volser)
- LNKLIST ADD NAME(listname) DSN=FEK.SFEKLOAD

以下のコンソール・コマンド・グループを使用して、動的に (次回の IPL までの)
LINKLIST 定義を作成できます。ここで volser は、マスター・カタログ内にカタ
ログされていないデータ・セットが存在するボリュームです。

1. LNKLIST DEFINE,NAME=LLTMP,COPYFROM=CURRENT
2. LNKLIST ADD NAME=LLTMP,DSN=FEK.SFEKAUTH,VOL=volser
3. LNKLIST ADD NAME=LLTMP,DSN=FEK.SFEKLOAD
4. LNKLIST ACTIVATE,NAME=LLTMP

必要な LINKLIST 定義と LPA 定義

リモート・システム・エクスプローラー (RSE) は、MVS ロード・ライブラリーへ
のアクセスを必要とする z/OS UNIX プロセスです。さらに、JES ジョブ・モニタ
ー、および統合デバッガーのサーバーもまた、システム、言語環境 (LE)、および C
ライブラリーへのアクセスを必要とします。以下の前提条件のライブラリーは、
STEPLIB または LINKLIST/LPALIB によって使用可能であることが必要です。

- システム・ロード・ライブラリー
 - SYS1.LINKLIB
- 言語環境プログラム・ランタイム
 - CEE.SCEERUN
 - CEE.SCEERUN2
- C++ の DLL クラス・ライブラリー
 - CBC.SCLBDLL
- ISPF の TSO/ISPF クライアント・ゲートウェイ
 - ISP.SISPLOAD

- ISP.SISPLPA
- REXX ランタイム・ライブラリー
 - REXX.*.SEAGLPA

オプションのサービスを使用できるようにするには、以下の追加ライブラリーを STEPLIB または LINKLIST/LPALIB を通じて使用可能にする必要があります。このリストには、Developer for System z と対話する製品 (IBM File Manager for z/OS など) に固有のデータ・セットは含まれていません。

- システム・ロード・ライブラリー (SSL 暗号化用)
 - SYS1.SIEALNKE
- システム・ロード・ライブラリー (統合デバッガー用)
 - SYS1.MIGLIB
- システム・ロード・ライブラリー (z/OS 1.13 以上の統合デバッガー用)
 - SYS1.SIEAMIGE
- システム・ロード・ライブラリー (Enterprise COBOL および PL/I の単体テスト用)
 - SYS1.CSSLIB
 - SYS1.SIXMLOD1

注:

- REXX 製品パッケージに代替ライブラリーを使用している場合、デフォルトの REXX ランタイム・ライブラリー名は REXX.*.SEAGALT であり、前のサンプルで使用した REXX.*.SEAGLPA ではありません。
- LPA にロードされるすべてのライブラリーは、自動的に、APF 許可が与えられていて、プログラム制御されるものとみなされます。これらのライブラリー用に、適切なセキュリティー管理を適切な場所に配置します。
- LPA 配置用に設計されたライブラリー (REXX.*.SEAGLPA など) は、LINKLIST または STEPLIB によってアクセスされる場合、追加のプログラム制御か APF 許可を必要とすることがあります。
- また、いくつかの相互前提条件となる製品 (IBM File Manager など) も STEPLIB または LINKLIST/LPALIB 定義を必要とします。詳細については、関連製品のカスタマイズ・ガイドを参照してください。

デフォルトでは、LINKLIST データ・セットは SYS1.PARMLIB(PROGxx) に定義します。LPA データ・セットは、SYS1.PARMLIB(LPALSTxx) で定義されます。

STEPLIB を使用する場合は、LINKLIST/LPALIB によって使用できないライブラリーを、rsed.envvars (RSE 構成ファイル) の STEPLIB ディレクティブで定義する必要があります。ただし、以下の点に注意してください。

- STEPLIB を z/OS UNIX で使用すると、パフォーマンスに悪い影響が出ます。
- 1 つの STEPLIB ライブラリーに APF 許可を与える場合、他のすべての STEPLIB ライブラリーにも APF 許可を与える必要があります。ライブラリーは、STEPLIB 内で許可のないライブラリーと混用した場合、APF 許可を失います。

- JCL で STEPLIB DD に追加されたライブラリーは、その JCL によって開始された z/OS UNIX プロセスに伝搬されません。

他の製品用の LINKLIST 定義

Developer for System z クライアントには、エンタープライズ・サービス・ツールと呼ばれるコード生成コンポーネントがあります。生成コードが診断用エラー・メッセージを発行するためには、FEK.SFEKLMOD ロード・ライブラリー内のすべての IRZM* モジュールと IIRZ* モジュールを、STEPLIB または LINKLIST を通じて使用できるようにする必要があります。

デフォルトでは、LINKLIST データ・セットは SYS1.PARMLIB(PROGxx) に定義します。

STEPLIB を使用する場合は、LINKLIST によって使用できないライブラリーを、コード (IMS またはバッチ・ジョブ) を実行するタスクの STEPLIB ディレクティブで定義する必要があります。ただし、1 つの STEPLIB ライブラリーに APF 許可を与える場合、他のすべての STEPLIB ライブラリーにも許可を与える必要があります。ライブラリーは、STEPLIB 内で許可のないライブラリーと混用した場合、APF 許可を失います。

PROCLIB の変更

このセクションでは、以下の PROCLIB の変更点について説明します。

- 『JMON、JES ジョブ・モニター開始タスク』
- 30 ページの『DBGMR、デバッグ・マネージャー開始タスク』
- 31 ページの『RSED、RSE デーモン開始タスク』
- 35 ページの『ELAXF* リモート・ビルド・プロシージャー』

以下のサブセクションに追加情報があります。

- 32 ページの『PARM 変数の JCL での制限』
- 33 ページの『TMPDIR 処理』
- 34 ページの『STDENV 処理』

次のセクションで示す開始タスク・プロシージャーおよびリモート・ビルド・プロシージャーは、使用する JES サブシステムに対して定義されたシステム・プロシージャー・ライブラリー内に存在する必要があります。次のセクションの説明では、IBM のデフォルトのプロシージャー・ライブラリー SYS1.PROCLIB を使用しています。

JMON、JES ジョブ・モニター開始タスク

サンプルの開始タスク・メンバー FEK.#CUST.PROCLIB(JMON) を、このメンバー内の記述に従ってカスタマイズし、SYS1.PROCLIB にコピーしてください。下記のコード・サンプルに示すように、以下の情報を提供します。

- (許可された) ロード・ライブラリー (デフォルトでは FEK) の高位修飾子
- JES ジョブ・モニター構成ファイル。デフォルトでは FEK.#CUST.PARMLIB (FEJJCNFG)

```

/*
/* JES JOB MONITOR
/*
//JMON      PROC PRM=,                * PRM='-TV' TO START TRACING
//          LEPRM='RPTOPTS(ON)',
//          HLQ=FEK,
//          CFG=FEK.#CUST.PARMLIB(FEJJCNFG)
/*
//JMON      EXEC PGM=FEJJMON,REGION=0M,TIME=NOLIMIT,
//          PARM=('&LEPRM,ENVAR("_CEE_ENVFILE_S=DD:ENVIRON")/&PRM')
//STEPLIB DD DISP=SHR,DSN=&HLQ..SFEKAUTH
//ENVIRON DD DISP=SHR,DSN=&CFG
//SYSPRINT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//          PEND
/*

```

図 1. JMON: JES ジョブ・モニター開始タスク

注:

- ・ 始動パラメーターの詳細については、221 ページの『第 11 章 オペレーター・コマンド』を参照してください。
- ・ サンプル JCL は、最初は FEK.SFEKSAMP(FEJJJCL) という名前ですが、20 ページの『カスタマイズのセットアップ』で FEK.#CUST.PROCLIB(JMON) という名前に変更されます。
- ・ トレースは、221 ページの『第 11 章 オペレーター・コマンド』で説明されているように、コンソール・コマンドによっても制御できます。
- ・ このタスクにおける推奨のワークロード・マネージャー (WLM) 目標については、「ホスト構成リファレンス」(SA88-4226) の『WLM に関する考慮事項』を参照してください。

DBGMGR、デバッグ・マネージャー開始タスク

サンプル開始タスク・メンバー FEK.#CUST.PROCLIB(DBGMGR) 内の説明に従ってこのメンバーをカスタマイズし、それを SYS1.PROCLIB にコピーします。下記のコード・サンプルに示すように、以下の情報を提供します。

- ・ タイムゾーン・オフセット (デフォルトは EST5DST)
- ・ 外部 (クライアント/ホスト) 通信に使用されるポート番号 (デフォルトは 5335)
- ・ 内部 (ホスト限定) 通信に使用されるポート (デフォルトは 5336)
- ・ ロード・ライブラリーの高位修飾子 (デフォルトは FEK)

```

/*
/* RDz Debug Manager
/*
/*DBGMGR  PROC PRM=, * PRM=DEBUG TO START TRACING
//          LEPRM='RPTOPTS(ON)',
//          TZ='EST5EDT',
//          CLIENT=5335,
//          HOST=5336,
//          HLQ=FEK
/*
/*DBGMGR  EXEC PGM=AQEZPCM,REGION=0M,TIME=NOLIMIT,
//          PARM=('&LEPRM ENVAR("TZ=&TZ")/&HOST &CLIENT 0 &PRM')
/*STEPLIB DD DISP=SHR,DSN=&HLQ..SFEKAUTH
/*SYSPRINT DD SYSOUT=*
/*SYSOUT DD SYSOUT=*
//          PEND
/*

```

図 2. DBGMGR: デバッグ・マネージャー開始タスク

注:

- これはオプションの開始タスクです。これは Developer for System z の統合デバッガー機能によって使用されます。詳しくは、144 ページの『(オプション) 統合デバッガー』を参照してください。
- サンプル JCL の最初の名前は FEK.SFEKSAMP(AQEZJCL) ですが、20 ページの『カスタマイズのセットアップ』で FEK.#CUST.PROCLIB(DBGMGR) に名前変更されます。
- このタスクにおける推奨のワークロード・マネージャー (WLM) 目標については、「ホスト構成リファレンス」(SA88-4226) の『WLM に関する考慮事項』を参照してください。

RSED、RSE デーモン開始タスク

サンプルの FEK.#CUST.PROCLIB(RSED) 開始タスク・メンバーを、このメンバー内の説明に従ってカスタマイズし、SYS1.PROCLIB にコピーしてください。下記のコード・サンプルに示すように、以下の情報を提供します。

- Developer for System z がインストールされているホーム・ディレクトリー、デフォルトは /usr/lpp/rdz。
- 構成ファイルのロケーション。デフォルトでは /etc/rdz

```

/*
/* RSE DAEMON
/*
/*RSED  PROC IVP=, * 'IVP' to do an IVP test
//          PORT=,
//          CNFG='/etc/rdz',
//          HOME='/usr/lpp/rdz'
/*
/*RSED  EXEC PGM=BPXBATSL,REGION=0M,TIME=NOLIMIT,
// PARM='PGM &HOME./bin/rsed.sh &IVP -C&CNFG -P&PORT'
//STDOUT DD SYSOUT=*
//STDERR DD SYSOUT=*
//          PEND
/*

```

図 3. RSED: RSE デーモン開始タスク

注:

- 始動パラメーターの詳細については、221 ページの『第 11 章 オペレーター・コマンド』を参照してください。
- サンプル JCL は、最初は FEK.SFEKSAMP(FEKRSED) という名前ですが、20 ページの『カスタマイズのセットアップ』で FEK.#CUST.PROCLIB(RSED) という名前に変更されます。
- ジョブ名の長さは、7 文字以下に制限してください。8 文字の名前を使用した場合、**modify** および **stop** オペレーター・コマンドが、「IEE342I MODIFY REJECTED-TASK BUSY」というメッセージを出して失敗します。この動作は、z/OS UNIX の子プロセスの設計によるものです。
- このタスク、および作成される子プロセスに関して推奨されるワークロード・マネージャー目標については、「ホスト構成リファレンス」(SA88-4226) の『WLM に関する考慮事項』を参照してください。子プロセスには、親タスクの名前 RSED にランダムな 1 桁の数字を付加した名前が付けられます (RSED8 など)。

PARM 変数の JCL での制限

PARM 変数の最大長は 100 文字であり、これが原因でカスタム・ディレクトリー名を使用する場合に問題が起きることもあります。この問題を回避するには、以下のオプションのいずれかを使用します。

- デフォルト値を使用する。

rsed.sh 始動スクリプトは、引数を指定せずに呼び出すことができ、その場合はデフォルトの引数値が使用されます。

- シンボリック・リンクを使用する。

長いディレクトリー名の省略方法として、シンボリック・リンクを使用できます。次のサンプルの z/OS UNIX コマンドは、別のディレクトリー (/long/directory/name/usr/lpp/rdz) へのシンボリック・リンク (/usr/lpp/rdz) を定義します。

```
ln -s /long/directory/name/usr/lpp/rdz /usr/lpp/rdz
```

- STDIN を使用します。

PARM フィールドが空の場合、**BPXBATCH** は z/OS UNIX シェルを始動し、STDIN から提供されたシェル・スクリプトを実行します。STDIN は、ORDONLY として割り当てられた z/OS UNIX ファイルでなければなりません。STDIN を使用することにより、TMPDIR などの PROC 変数が使用不可になります。このシェルは、/etc/profile および \$HOME/.profile シェル・ログオン・スクリプトを実行します。

この方式を使用するには、最初に始動 JCL を更新して、以下のサンプルのようなものにします。

```

/*
/* RSE DAEMON - USING STDIN
/*
//RSED      PROC CNFG='/etc/rdz'
/*
//RSE      EXEC PGM=BPXBATCH,REGION=0M,TIME=NOLIMIT
//STDOUT   DD SYSOUT=*
//STDERR   DD SYSOUT=*
//STDIN    DD PATHOPTS=(ORDONLY),PATH='&CNFG./rsed.stdin.sh'
//          PEND
//
/*

```

図 4. RSED - 代替の RSE デーモンの始動

次に、RSE デーモンを始動するシェル・スクリプト (この例では /etc/rdz/rsed.stdin.sh) を作成します。このファイルは、TSO **OEDIT** コマンドで編集できます。このスクリプトの内容は、次のサンプルのようになります。

```

CNFG=/etc/rdz
PORT=
IVP=
/long/directory/name/usr/lpp/rdz/bin/rsed.sh $IVP -C$CNFG -P$PORT -T$TMPDIR

```

図 5. rsed.stdin.sh: 代替 RSE デーモンの始動

注: この方式を使用する場合、RSE デーモン自体は、RSEDx アドレス・スペースを除き、RSED アドレス・スペースではアクティブになりません。これは、z/OS UNIX では子プロセス (シェルの始動など) を別々のアドレス・スペースで実行するからです。_BPX_SHAREAS=YES ディレクティブを指定して STDENV DD を追加しても、このディレクティブは解釈されるのが遅すぎるため、この動作は変更されません。この副次作用のため、Developer for System z オペレーター・コマンドの使用が非常に複雑になります。

TMPDIR 処理

z/OS UNIX は、開始タスクの始動時に特定のコマンドの処理を可能にするために、/tmp、または TMPDIR 変数で参照される別のディレクトリーへの書き込みアクセスを必要とします。Developer for System z は、開始タスクの始動時に、以下のロジックに従って TMPDIR を設定します。

開始タスクの始動時に、Developer for System z は TMPDIR が既に設定 (DD STDENV) されているかどうか検査します。設定されている場合、開始タスクはその値を使用します。TMPDIR が設定されていない場合、開始タスクは、/tmp が使用可能かどうかを検査します。使用できない場合、開始タスクは、開始タスクのユーザー ID に割り当てられているホーム・ディレクトリーが使用可能かどうかを検査します。このディレクトリーも使用できない場合は、始動が失敗します。

/tmp のデフォルトのバックアップであるホーム・ディレクトリーを使用できない場合は、以下のサンプルに示すように、DD STDENV を使用して TMPDIR を事前に定義する必要があります。


```

/*
/* RSE DAEMON
/*
//RSED      PROC IVP=,                * 'IVP' to do an IVP test
//          PORT=,
//          CNFG='/etc/rdz',
//          HOME='/usr/lpp/rdz'
/*
//RSED      EXEC PGM=BPXBATSL,REGION=0M,TIME=NOLIMIT,
// PARM='PGM &HOME./bin/rsed.sh &IVP -C&CNFG -P&PORT'
//STDOUT    DD SYSOUT=*
//STDERR    DD SYSOUT=*
//STDENV    DD PATHOPTS=(ORDONLY),PATH='&CNFG./rsed.stdenv'
//          PEND
/*

```

図 6. RSED: 代替 TMPDIR 処理

次に、TMPDIR 定義を保持するファイル (この例では /etc/rdz/rsed.stdenv) を作成します。このファイルは、TSO **OEDIT** コマンドで編集できます。このファイルの内容は、次のサンプルのようになります。

```
TMPDIR=/tmp
```

図 7. rsed.stdenv: 代替 TMPDIR 処理

rsed.envvars に TMPDIR 変数を指定すると、この変数は、開始タスクが rsed.envvars を解釈できるようになるとすぐに使用されますが、その場合であっても、rsed.envvars を DD STDENV にリンクしないでください。これを行うと始動に失敗するためです。

STDENV 処理

Developer for System z では、rsed.envvars で使用可能な変数を RSED 開始タスクの DD STDENV で指定できます。この変数は RSE スレッド・プールに伝搬されるため、すべてのエンド・ユーザーにも伝搬されます。

例えば、これにより、ログ・ディレクトリーのような固有でなければならない情報を DD STDENV に指定できるので、複数の RSED 開始タスクに対して単一の rsed.envvars 構成ファイルを使用できます。35 ページの図 8 には DD STDENV を使用する RSED 開始タスクを示し、35 ページの図 9 には DD STDENV で定義された変数を使用する rsed.envvars 構成ファイルを示します。

/tmp のデフォルトのバックアップであるホーム・ディレクトリーを使用できない場合は、以下のサンプルに示すように、DD STDENV を使用して TMPDIR を事前に定義する必要があります。


```

/**
/** RSE DAEMON
/**
//RSED      PROC IVP=,                * 'IVP' to do an IVP test
//          PORT=,
//          CNFG='/etc/rdz',
//          HOME='/usr/lpp/rdz'
/**
//RSED      EXEC PGM=BPXBATSL,REGION=0M,TIME=NOLIMIT,
// PARM='PGM &HOME./bin/rsed.sh &IVP -C&CNFG -P&PORT'
//STDOUT    DD SYSOUT=*
//STDERR    DD SYSOUT=*
//STDENV    DD *
MYSYSTEM=CDFMVS08
//          PEND
/**

```

図 8. RSED: STDENV 処理

```
_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Ddaemon.log=/var/rdz/logs/$MYSYSTEM"
```

図 9. rsed,envvars: STDENV 処理

注: z/OS UNIX には、DD STDENV 用の制限付き構文解析機能があります。コメントおよび variable=value ディレクティブのみが使用可能です。

variable=\$other-variable はサポートされていません。US コード・ページを使用する場合、コメント行は番号記号 (#) で始まります。

ELAXF* リモート・ビルド・プロシージャ

Developer for System z は、CICS BMS マップ、IMS MFS 画面、および COBOL、PL/I、アセンブラー、C/C++ の各プログラムの、JCL 生成、リモート・プロジェクト・ビルド、およびリモート構文検査の各フィーチャーに使用できるサンプル JCL プロシージャを提供します。これらのプロシージャを使用すると、インストールごとに独自の標準を適用でき、開発者は、同じプロシージャを同じコンパイラー・オプションおよびコンパイラー・レベルで使用できます。

サンプル・プロシージャとその機能を、表 8 に示します。

表 8. サンプル ELAXF* プロシージャ

メンバー	目的
ELAXFADT	高水準アセンブラー・プログラムのアセンブルとデバッグのためのサンプル・プロシージャ。
ELAXFASM	高水準アセンブラー・プログラムのアセンブルのためのサンプル・プロシージャ。
ELAXFBMS	CICS BMS オブジェクトおよびそれに対応する copy、dsect、または include メンバーを作成するためのサンプル・プロシージャ。
ELAXFCOC	COBOL コンパイル、統合 CICS 変換、および統合 DB2 変換を行うためのサンプル・プロシージャ。
ELAXFCOP	COBOL プログラムに埋め込まれた EXEC SQL ステートメントの DB2 プリプロセスのためのサンプル・プロシージャ。
ELAXFCOT	COBOL プログラムに組み込まれている EXEC CICS ステートメントの CICS 変換するためのサンプル・プロシージャ。
ELAXFCPC	C コンパイルのためのサンプル・プロシージャ。

表 8. サンプル ELAXF* プロシージャ (続き)

メンバー	目的
ELAXFCPP	C++ コンパイルのためのサンプル・プロシージャ。
ELAXFCPI	SCM プリプロセッサ・ステートメント (-INC および ++INCLUDE) を使用した COBOL コンパイルのためのサンプル・プロシージャ。
ELAXFDCL	プログラムを TSO モードで実行するためのサンプル・プロシージャ。
ELAXFGO	GO ステップのためのサンプル・プロシージャ。
ELAXFLNK	C/C++、COBOL、PLI、および高水準アセンブラの各プログラムをリンクするためのサンプル・プロシージャ。
ELAXFMFS	IMS MFS 画面を作成するためのサンプル・プロシージャ。
ELAXFPLP	PLI プログラムに埋め込まれた EXEC SQL ステートメントの DB2 プリプロセスのためのサンプル・プロシージャ。
ELAXFPLT	PLI プログラムに埋め込まれた EXEC CICS ステートメントの CICS 変換を行うためのサンプル・プロシージャ。
ELAXFPLI	PL/I コンパイル、統合 CICS 変換、および統合 DB2 変換を行うためのサンプル・プロシージャ。
ELAXFPP1	SCM プリプロセッサ・ステートメント (-INC および ++INCLUDE) を含む PL/I コンパイルのためのサンプル・プロシージャ。
ELAXFSP	ストアード・プロシージャを DB2 に登録するためのサンプル・プロシージャ。
ELAXFSQL	SQL を呼び出すためのサンプル・プロシージャ。
ELAXFTSO	生成された DB2 コードを TSO モードで実行およびデバッグするためのサンプル・プロシージャ。
ELAXFUOP	CICS または IMS サブシステムで実行するプログラムをビルドするとき、UOPT ステップを生成するためのサンプル・プロシージャ。

プロシージャの名前とプロシージャ内のステップの名前は、Developer for System z クライアントに付属しているデフォルト・プロパティに一致しています。プロシージャの名前またはプロシージャ内のステップの名前を変更する場合は、すべてのクライアント上の対応するプロパティ・ファイルを更新する必要があります。プロシージャ名とステップ名は変更しないでください。

サンプルのビルド・プロシージャ・メンバー FEK.#CUST.PROCLIB(ELAXF*) を、各メンバー内で説明されているようにカスタマイズし、SYS1.PROCLIB にコピーします。各種製品ライブラリーに、表 9 に示した正しい高位修飾子を指定します。

表 9. ELAXF* 高位修飾子チェックリスト

製品	デフォルト HLQ	値
Developer for System z	FEK	
CICS	CICSTS42.CICS	
DB2	DSNA10	
IMS	IMS	
COBOL	IGY.V4R2M0	
PL/I	PLI.V4R2M0	
C/C++	CBC	

表 9. ELAXF* 高位修飾子チェックリスト (続き)

製品	デフォルト HLQ	値
LE	CEE	
システム LINKLIB	SYS1	
システム MACLIB	SYS1	

一部の ELAXF* プロシージャは、固定された低位修飾子を持たないデータ・セット名を参照します。その例の 1 つは DB2 実行ライブラリーです。これは、DB2 管理者によってコンパイルされた DB2 ユーティリティを保持します。表 10 を使用して、デフォルトのデータ・セット名をユーザーのサイトで使用されている名前にマップします。

表 10. ELAXF* : 完全修飾データ・セット・チェックリスト

製品	デフォルト DSN	値
Developer for System z - SQL サンプル	FEK.#CUST.SQL	
DB2 実行ライブラリー	DSNA10.RUNLIB.LOAD	

ELAXF* プロシージャをシステム・プロシージャ・ライブラリーにコピーできない場合、JCLLIB カード (JOB カードの直後) をクライアントのジョブ・プロパティに追加するように Developer for System z ユーザーに依頼します。

```
//MYJOB    JOB <job parameters>
//PROCS    JCLLIB ORDER=(FEK.#CUST.PROCLIB)
```

セキュリティ定義

Developer for System z のセキュリティ定義を作成するには、サンプルの FEKRACF メンバーをカスタマイズして実行依頼します。このジョブを実行依頼するユーザーは、RACF SPECIAL などのセキュリティ管理者特権を持っている必要があります。

FEKRACF は FEK.#CUST.JCL に置かれます。ただし、FEK.SFEKSAMP(FEKSETUP) ジョブをカスタマイズして実行依頼したときに、別のロケーションを指定した場合は除きます。詳しくは、20 ページの『カスタマイズのセットアップ』を参照してください。

注:

- CA ACF2™ for z/OS を使用しているサイトの場合は、CA サポート・サイト (<https://support.ca.com>) の製品ページを参照し、関連する Developer for System z Knowledge Document TEC492389 を確認してください。この Knowledge Document には、Developer for System z を正しく構成するために必要なセキュリティ・コマンドの詳細が記載されています。
- CA Top Secret® for z/OS を使用しているサイトの場合は、CA サポート・サイト (<https://support.ca.com>) の製品ページを参照し、関連する Developer for System z Knowledge Document TEC492091 を確認してください。この Knowledge Document には、Developer for System z を正しく構成するために必要なセキュリティ・コマンドの詳細が記載されています。

以下にリストする Developer for System z のセキュリティー関連定義については、171 ページの『第 9 章 セキュリティー定義』で詳しく説明します。

- セキュリティーの設定およびクラスをアクティブにする
- Developer for System z ユーザーの OMVS セグメントを定義する
- Developer for System z 開始タスクの定義
- セキュアな z/OS UNIX サーバーとして RSE を定義する
- RSE の MVS プログラム制御ライブラリーを定義する
- RSE の PassTicket サポートを定義する
- RSE のアプリケーション・セキュリティーを定義する
- JES コマンド・セキュリティーを定義する
- 統合デバッガーへのアクセスの定義
- データ・セット・プロファイルを定義する

重要: PassTicket が正しくセットアップされていないと、クライアントの接続要求は失敗します。

FEJJCNFG、JES ジョブ・モニター構成ファイル

JES ジョブ・モニター (JMON) は、JES に関連したすべてのサービスを提供します。JES ジョブ・モニターの動作は、FEJJCNFG 内の定義によって制御できます。

FEJJCNFG は FEK.#CUST.PARMLIB に置かれます。ただし、FEK.SFEKSAMP (FEKSETUP) ジョブをカスタマイズして実行依頼したときに、別のロケーションを指定した場合は除きます。詳しくは、20 ページの『カスタマイズのセットアップ』を参照してください。

サンプルの JES ジョブ・モニター構成メンバー、FEJJCNFG を、次の例に示すようにカスタマイズしてください。US コード・ページを使用する場合、コメント行は番号記号 (#) で始まります。データ行には、ディレクティブとそれに割り当てられる値のみを入れることができます。同じ行にコメントを入れることはできません。

注: 変更を有効にするには、JMON 開始タスクを再始動する必要があります。

```

SERV_PORT=6715
TZ=EST5EDT
#APPLID=FEKAPPL
#AUTHMETHOD=SAF
#CODEPAGE=UTF-8
#CONCHAR=$
#CONSOLE_NAME=JMON
#GEN_CONSOLE_NAME=OFF
#HOST_CODEPAGE=IBM-1047
#LIMIT_COMMANDS=NOLIMIT
#LIMIT_CONSOLE=LIMITED
#LIMIT_VIEW=USERID
#LISTEN_QUEUE_LENGTH=5
#LOOPBACK_ONLY=ON
#MAX_DATASETS=32
#MAX_THREADS=200
#TIMEOUT=3600
#TIMEOUT_INTERVAL=1200
#TRACE_STORAGE=OFF
#SEARCHALL=OFF
#SUBMIT_TIMEOUT=30
#SUBMITMETHOD=TSO
#TSO_TEMPLATE=FEK.#CUST.CNTL(FEJTS0)

```

図 10. FEJJCNFG、JES ジョブ・モニター構成ファイル

SERV_PORT

JES ジョブ・モニターのポート番号。デフォルト・ポートは 6715 です。必要であればポートを変更できます。

注:

- この値は、rsed.envvars 構成ファイル内で JES ジョブ・モニターに設定したポート番号に一致する必要があります。これらの値が異なる場合、RSE はクライアントを JES ジョブ・モニターに接続できません。RSE に対して変数を定義する方法については、43 ページの『rsed.envvars、RSE 構成ファイル』を参照してください。
- ポートを選択する前に、TSO コマンド **NETSTAT** および **NETSTAT PORTL** を使用して、そのポートがシステム上で使用可能であることを確認してください。

TZ タイム・ゾーン・セレクター。デフォルトは EST5EDT です。デフォルトのタイム・ゾーンは、UTC -5 時間 (東部標準時 (EST) 東部夏時間 (EDT)) です。この値は、使用するタイム・ゾーンを表すように変更してください。詳細は、「UNIX System Services コマンド解説書」(SA88-8641) に記載されています。

以下の定義はオプションです。省略した場合は、以下に示すデフォルト値が使用されます。

APPLID

セキュリティ・ソフトウェアに対して JES ジョブ・モニターを識別するために使用するアプリケーション ID を指定します。デフォルトは FEKAPPL です。コメントを外し、必要なアプリケーション ID に変更してください。

注: この値は、rsed.envvars 構成ファイル内で RSE に設定したアプリケーション ID に一致する必要があります。これらの値が異なる場合、RSE は

クライアントを JES ジョブ・モニターに接続できません。RSE に対して変数を定義する方法については、43 ページの『rsed.envvars、RSE 構成ファイル』を参照してください。

AUTHMETHOD

デフォルトは SAF で、System Authorization Facility (SAF) セキュリティー・インターフェースが使用されることを意味します。IBM サポートから変更するように指示された場合以外は、これを変更しないでください。

CODEPAGE

ワークステーションのコード・ページです。デフォルトは UTF-8 です。ワークステーションのコード・ページは UTF-8 に設定されており、通常、これを変更するべきではありません。通貨記号など、マルチリンガル文字の扱いに問題がある場合は、ディレクティブをコメント解除し、ワークステーションのコード・ページに一致するように UTF-8 を変更すると良いでしょう。

CONCHAR

JES コンソール・コマンド文字を指定します。CONCHAR のデフォルトは、JES2 の場合は CONCHAR=\$、JES3 の場合は CONCHAR=* です。コメント解除し、要求されたコマンド文字に変更してください。

CONSOLE_NAME

ジョブに対するコマンド（「保留」、「保留解除」、「キャンセル」、および「パージ」）を発行するために使用する、EMCS コンソールの名前を指定します。デフォルトは JMON です。コメントを外し、以下のガイドラインを使用して、必要なコンソール名に変更してください。

- CONSOLE_NAME は、2 から 8 文字の英数字のコンソール名か、&SYSUID（引用符は付けない）でなければなりません。
- コンソール名を指定した場合、その名前で単一のコンソールがすべてのユーザーによって使用されます。その名前のコンソールが既に使用されている場合、クライアントによって発行されたコマンドは失敗します。
- &SYSUID を指定した場合は、クライアント・ユーザー ID がコンソール名として使用されます。このため、ユーザーごとに異なるコンソールが使用されます。その名前のコンソールが既に使用されている場合（例えば、ユーザーが SDSF ULOG を使用している場合）、クライアントにより発行されたコマンドが失敗するかどうかは、GEN_CONSOLE_NAME の設定によって決まります。

どのようなコンソール名を使用しても、コマンドを要求しているクライアントのユーザー ID がコンソールの LU として使用され、トレースでは syslog メッセージ IEA630I および IEA631I が残されます。

```
IEA630I OPERATOR console NOW ACTIVE,  SYSTEM=sysid, LU=id
IEA631I OPERATOR console NOW INACTIVE, SYSTEM=sysid, LU=id
```

GEN_CONSOLE_NAME

代替コンソール名の自動生成を使用可能または使用不可にします。デフォルトは OFF です。代替コンソール名を使用可能にするには、コメントを外して ON に変更します。

このディレクティブを使用するのは、CONSOLE_NAME が &SYSUID と等しく、かつユーザー ID をコンソール名として使用できない場合に限られます。

GEN_CONSOLE_NAME=ON の場合は、ユーザー ID に数字を 1 つだけ付加することによって、代替コンソール名が生成されます。数字 0 から 9 までが試みられます。使用可能なコンソールがない場合、クライアントが発行したコマンドは失敗します。

GEN_CONSOLE_NAME=OFF の場合、クライアントが発行したコマンドは失敗します。

注: 有効な設定値は、ON と OFF だけです。

HOST_CODEPAGE

ホスト・システムのコード・ページ。デフォルトは IBM-1047 です。コメント解除し、ご使用のホスト・システム・コード・ページに合わせて変更してください。

このコード・ページはデータ変換処理には使用されず、サーバーの操作およびクライアント接続のセットアップにのみ使用されることに注意してください。データ変換処理に使用するコード・ページは、Developer for System z クライアントから指定されます (これは、「MVS ファイル」サブシステムのプロパティーから取得されます)。

LIMIT_COMMANDS

ユーザーがどのジョブに対して、選択された JES コマンド (「JCL の表示」、「保留」、「保留解除」、「キャンセル」、および「ページ」) を発行できるかを定義します。デフォルト (LIMIT_COMMANDS=USERID) では、コマンドの発行対象は、そのユーザーが所有するジョブだけに制限されます。セキュリティ製品による許可がある場合に、ユーザーがすべてのスプール・ファイルに対してコマンドを発行できるようにするには、このディレクティブをコメントを外し、LIMITED または NOLIMIT を指定します。

表 11. LIMIT_COMMANDS コマンドの許可のマトリックス

	ジョブ所有者	
LIMIT_COMMANDS	ユーザー	その他
USERID (デフォルト)	許可される	許可されない
LIMITED	許可される	セキュリティ・プロファイルによって明示的に許可された場合にのみ許可される
NOLIMIT	許可される	セキュリティ・プロファイルによって許可された場合、または JESSPOOL クラスがアクティブでない場合は許可される

注: 有効な設定値は、USERID、LIMITED、および NOLIMIT だけです。

LIMIT_CONSOLE

サポートされている JES コマンド (「保留」、「保留解除」、「キャンセル」、および「ページ」) を実行するために使用するコンソールに対して、どの程度の権限を付与するかを定義します。デフォルト (LIMIT_CONSOLE=LIMITED) では、権限は、OPERCMDS クラスのセキュリティ・プロファイルによって保護されているコマンドに制限されます。セキュ

リティー・プロファイルによって保護されていない、サポートされている JES コマンドを実行できるようにするには、このディレクティブのコメントを外して NOLIMIT を指定します。

コマンド用のセキュリティ・プロファイルが存在する場合には、LIMIT_CONSOLE の設定に関係なく、ユーザーにコマンドを実行するための十分な許可が必要です。有効な設定値は、LIMITED と NOLIMIT だけです。

LIMIT_VIEW

ユーザーが表示できる出力を定義します。デフォルト (LIMIT_VIEW=NOLIMIT) では、ユーザーはセキュリティ製品による許可があれば、すべての JES 出力を表示できます。ユーザーが所有する出力に表示を限定するには、このディレクティブのコメントを外して USERID を指定します。

注: 有効な設定値は、USERID と NOLIMIT だけです。

LISTEN_QUEUE_LENGTH

TCP/IP listen キュー長。デフォルトは 5 です。IBM サポートから変更するように指示された場合以外は、これを変更しないでください。

LOOPBACK_ONLY

JES ジョブ・モニターをループバック・アドレスにのみバインドするか、使用可能なすべての TCP/IP スタックにバインドするかを定義します。ループバックへのバインドのほうが安全です。この z/OS システムにローカルなタスクのみが JES ジョブ・モニターに接続できるようになるからです。デフォルトは ON です。JES ジョブ・モニターをすべての TCP/IP スタックにバインドするには、このディレクティブのコメントをはずして OFF を指定します。

MAX_DATASETS

JES ジョブ・モニターがクライアント (例えば、SYSOUT、SYSPRINT、SYS00001 など) に返すスプール出力データ・セットの最大数。デフォルトは 32 です。最大値は 2147483647 です。

MAX_THREADS

1 つの JES ジョブ・モニターを一度に使用できるユーザーの最大数。デフォルトは 200 です。最大値は 2147483647 です。この数を大きくすると、JES ジョブ・モニターのアドレス・スペースのサイズを大きくしなければならない場合があります。

TIMEOUT

クライアントとの対話がないためにスレッドが強制終了されるまでの時間の長さ (秒単位)。デフォルトは 3600 (1 時間) です。最大値は 2147483647 です。TIMEOUT=0 は、この機能を無効にします。

TIMEOUT_INTERVAL

タイムアウト検査の間隔を表す秒数。デフォルトは 1200 です。最大値は 2147483647 です。

TRACE_STORAGE

ストレージのトレースを有効にします。デフォルトは OFF です。有効な値は、ON と OFF だけです。IBM サポートから指示された場合にのみ使用してください。各コマンドの後にストレージ・レポートを DD SYSOUT に書き込むには、このディレクティブのコメントを外して ON を指定します。

SEARCHALL

JES ジョブ・モニター・フィルターに一致する APPC および z/OS UNIX 出力 (例えば、CRASTART メソッドを使用して始動した Developer for System z CARMA サーバーによって SYSOUT に書き込まれた出力) を収集します。デフォルトは OFF です。有効な値は、ON と OFF だけです。追加のスプール・ファイルを収集するには、このディレクティブのコメントを外して ON を指定します。

SUBMIT_TIMEOUT

TSO_TEMPLATE ジョブが完了するまで Developer for System z が待機する秒数。デフォルトは 30 です。最大値は 2147483647 です。注:
SUBMIT_TIMEOUT は、SUBMITMETHOD=TSO も指定されていないと効果がありません。

SUBMITMETHOD=TSO

TSO によるジョブの実行依頼。デフォルト (SUBMITMETHOD=JES) では、ジョブが JES へ直接実行依頼されます。TSO **SUBMIT** コマンドを使用してジョブを実行依頼するには、このディレクティブのコメントを外して TSO を指定します。この方式を使用すると TSO 出口を呼び出すことができますが、パフォーマンスが低下します。

注:

- 有効な設定値は、TSO と JES だけです。
- SUBMITMETHOD=TSO を指定した場合は、TSO_TEMPLATE も定義する必要があります。

TSO_TEMPLATE

TSO によってジョブを実行依頼するためのラッパー JCL。デフォルト値は FEK.#CUST.CNTL(FEJTSO) です。このステートメントは、TSO **SUBMIT** コマンドのラッパーとして使用される JCL の完全修飾メンバー名を参照します。詳細については、SUBMITMETHOD ステートメントを参照してください。

注:

- サンプルのラッパー・ジョブは FEK.#CUST.CNTL(FEJTSO) で提供されています。必要なカスタマイズの詳細については、このメンバーを参照してください。
- TSO_TEMPLATE は、SUBMITMETHOD=TSO も指定されていないと効果がありません。

rsed.envvars、RSE 構成ファイル

RSE サーバー・プロセス (RSE デーモン、RSE スレッド・プール、および RSE サーバー) は、rsed.envvars 内の定義を使用します。オプションの Developer for System z サービスおよびサード・パーティー・サービスも、この構成ファイルを使用して、使用する環境変数を定義することができます。

リモート・システム・エクスプローラー (RSE) は、クライアントをホスト・システムに接続したり、特定のサービス用に他のサーバーを始動するなどの、コア・サービスを提供します。

`rsed.envvars` は `/etc/rdz/` に置かれます。ただし、`FEK.SFEKSAMP(FEKSETUP)` ジョブをカスタマイズして実行依頼したときに、別のロケーションを指定した場合は除きます。詳しくは、20 ページの『カスタマイズのセットアップ』を参照してください。このファイルは、**TSO OEDIT** コマンドで編集できます。

以下のサンプルの `rsed.envvars` ファイルを参照してください。このファイルは、使用するシステム環境に合わせてカスタマイズする必要があります。US コード・ページを使用する場合、コメント行は番号記号 (#) で始まります。データ行には、ディレクティブとそれに割り当てられる値のみを入れることができ、その同じ行にコメントを入れることはできません。行の継続および等号 (=) の前後のスペースはサポートされていません。

注: 変更を有効にするには、**RSED** 開始タスクを再始動する必要があります。

```

#####
# (1) required definitions
JAVA_HOME=/usr/lpp/java/J6.0
RSE_HOME=/usr/lpp/rdz
_RSE_RSED_PORT=4035
_RSE_JMON_PORT=6715
RSE_HLQ=FEK
_RSE_HOST_CODEPAGE=IBM-1047
TZ=EST5EDT
LANG=C
PATH=/bin:/usr/sbin
_CEE_DMPRTARG=/tmp
STEPLIB=NONE
#STEPLIB=$STEPLIB:CEE.SCEERUN:CEE.SCEERUN2:CBC.SCLBDLL
_RSE_JAVAOPTS=""
_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Xms128m -Xmx512m"
_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Ddaemon.log=/var/rdz/logs"
_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Duser.log=/var/rdz/logs"
_RSE_JAVAOPTS="$_RSE_JAVAOPTS -DDSTORE_LOG_DIRECTORY="
_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Dlog.retention.period=5"
#_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Dmaximum.clients=30"
#_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Dmaximum.threads=520"
#_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Dminimum.threadpool.process=1"
#_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Dmaximum.threadpool.process=100"
#_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Dipv6=true"
#_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Denable.dvIPA=true"
#_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Ddisplay.users=true"
#_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Dkeep.all.logs=false"
#_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Dkeep.last.log=true"
#_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Denable.standard.log=true"
#_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Dlog.mode=RW.N.N"
#_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Dlog.secure.mode=false"
#_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Denable.port.of.entry=true"
#_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Denable.certificate.mapping=false"
#_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Denable.automount=true"
#_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Denable.audit.log=true"
#_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Daudit.cycle=30"
#_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Daudit.retention.period=0"
#_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Daudit.log.mode=RW.R.N"
#_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Daudit.action=<user_exit>"
#_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Daudit.action.id=<userid>"
#_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Dlogon.action=<user_exit>"
#_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Dlogon.action.id=<userid>"
#_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Ddeny.nonzero.port=true"
#_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Dsingle.logon=false"
#_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Dprocess.cleanup.interval=0"
#_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Dreject.logon.threshold=1000000"
#_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Dinclude.c=/etc/rdz/include.conf"
#_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Dinclude.cpp=/etc/rdz/include.conf"
#_RSE_JAVAOPTS="$_RSE_JAVAOPTS -DCPP_CLEANUP_INTERVAL=60000"
#_RSE_JAVAOPTS="$_RSE_JAVAOPTS -DRIS_BUFFER=8"
#_RSE_JAVAOPTS="$_RSE_JAVAOPTS -DAPPLID=FEKAPPL"
#_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Dkeep.stats.copy.local=true"
#_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Ddebug.miner.autoreconnect=0"
#_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Ddebug.miner.localhost=localhost"
#_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Dsearch.server.limit.hits=0"
#_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Dsearch.server.limit.scanned_objects=0"
#_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Dsearch.server.limit.lines=0"
#_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Dsearch.server.limit.timeout=0"
#_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Dsearch.server.limit.errcount=true"
#_RSE_JAVAOPTS="$_RSE_JAVAOPTS -DDISABLE_TEXT_SEARCH=true"
#_RSE_JAVAOPTS="$_RSE_JAVAOPTS -DDENY_PASSWORD_SAVE=true"

```

図 11. rsed.envvars: RSE 構成ファイル

```

# RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DHIDE_ZOS_UNIX=true"
# RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DDISABLE_REMOTE_INDEX_SEARCH=true"
# RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DDISABLE_DELETE_IN_SUBPROJECT=true"
# RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DDENY_PASSWORD_SAVE=true"
# RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DDISABLE_REMOTE_INDEX_SEARCH=true"
# RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DDISABLE_DELETE_IN_SUBPROJECT=true"
# RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DDENY_PASSWORD_SAVE=true"
# RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DHIDE_ZOS_UNIX=true"
# RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DDSTORE_IDLE_SHUTDOWN_TIMEOUT=3600000"
# RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DDSTORE_USE_THREADED_MINERS=false"
# RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DDSTORE_SSL_ALGORITHM=TLSv1.2"
# RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DDSTORE_TCP_NO_DELAY=true"
# RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DDSTORE_TRACING_ON=true"
# RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DDSTORE_MEMLOGGING_ON=true"
# =====
# (2) required definitions for TSO/ISPF Client Gateway
CGI_ISPHOME=/usr/lpp/ispf
CGI_ISPCONF=/etc/rdz
CGI_ISPWORK=/var/rdz
#STEPLIB=$STEPLIB:ISP.SISPLoad:ISP.SISPLPA:SYS1.LINKLIB
_RSE_ISPF_OPTS=""
# RSE_ISPF_OPTS="$ _RSE_ISPF_OPTS&ISPPROF=&SYSUID..ISPPROF"
#CGI_ISPPREF="&SYSPREF..ISPF.VCMISPF"
# =====
# (3) required definitions for SCLM Developer Toolkit
_SCLMDT_CONF_HOME=/var/rdz/sclmdt
#STEPLIB=$STEPLIB:$RSE_HLQ.SFEKAUTH:$RSE_HLQ.SFEKLOAD
# _SCLMDT_TRANTABLE=FEK.#CUST.LSTRANS.FILE
#ANT_HOME=/usr/lpp/Apache/Ant/apache-ant-1.7.1
# =====
# (4) optional definitions
# RSE_PORTRANGE=8108-8118
# BPXK_SETIBMOPT_TRANSPORT=TCPIP
#TMPDIR=/tmp
# RSE_FEK_SAF_CLASS=FACILITY
# RSE_LDAP_SERVER=ldap_server_url
# RSE_LDAP_PORT=389
# RSE_LDAP_PTC_GROUP_SUFFIX="o=PTC,c=DeveloperForZ"
#GSK_CRL_SECURITY_LEVEL=HIGH
#GSK_LDAP_SERVER=ldap_server_url
#GSK_LDAP_PORT=ldap_server_port
#GSK_LDAP_USER=ldap_userid
#GSK_LDAP_PASSWORD=ldap_server_password
#STEPLIB=$RSE_HLQ.SFEKLOAD:SYS1.CSSLIB:SYS1.SIXMLOD1
#RSE_UBLD_DD=$CGI_ISPCONF/ISPF.conf
#RSE_UBLD_STEPLIB=$STEPLIB
# =====
# (5) do not change unless directed by IBM support center
_RSE_SAF_CLASS=/usr/include/java_classes/IRRacf.jar
_CEE_RUNOPTS="ALL31(ON) HEAP(32M,32K,ANYWHERE,KEEP,,) TRAP(ON)"
_BPX_SHAREAS=YES
_BPX_SPAWN_SCRIPT=YES
_EDC_ADD_ERRNO2=1
JAVA_PROPAGATE=NO
RSE_DSN_SFEKLOAD=$RSE_HLQ.SFEKLOAD
RSE_LIB=$RSE_HOME/lib
PATH=.:$JAVA_HOME/bin:$RSE_HOME/bin:$CGI_ISPHOME/bin:$PATH
LIBPATH=$JAVA_HOME/bin:$JAVA_HOME/bin/classic:$RSE_LIB:$RSE_LIB/icuc
LIBPATH=.:/usr/lib:$LIBPATH
CLASSPATH=$RSE_LIB:$RSE_LIB/dstore_core.jar:$RSE_LIB/clientserver.jar
CLASSPATH=$CLASSPATH:$RSE_LIB/dstore_extra_server.jar
CLASSPATH=$CLASSPATH:$RSE_LIB/zosserver.jar

```

図 12. rsed.envvars: RSE 構成ファイル (続き)

```

CLASSPATH=$CLASSPATH:$RSE_LIB/dstore_miners.jar
CLASSPATH=$CLASSPATH:$RSE_LIB/universalminers.jar:$RSE_LIB/mvsminers.jar
CLASSPATH=$CLASSPATH:$RSE_LIB/carma.jar:$RSE_LIB/luceneminer.jar
CLASSPATH=$CLASSPATH:$RSE_LIB/mvsluceneminer.jar:$RSE_LIB/cdzminer.jar
CLASSPATH=$CLASSPATH:$RSE_LIB/mvscdzminer.jar:$RSE_LIB/jesminers.jar
CLASSPATH=$CLASSPATH:$RSE_LIB/debug_miner.jar
CLASSPATH=$CLASSPATH:$RSE_LIB/mvsutil.jar:$RSE_LIB/jesutils.jar
CLASSPATH=$CLASSPATH:$RSE_LIB/lucene-core-2.3.2.jar
CLASSPATH=$CLASSPATH:$RSE_LIB/cdtparser.jar:$RSE_LIB/wdzBidi.jar
CLASSPATH=$CLASSPATH:$RSE_SAF_CLASS
CLASSPATH=.:$CLASSPATH
_RSE_PTC=$RSE_LDAP_PTC_GROUP_SUFFIX
_RSE_ISPF_OPTS="&SESSION=SPAWN$_RSE_ISPF_OPTS"
_RSE_JAVAOPTS="$RSE_JAVAOPTS -Dldap.server.address=$_RSE_LDAP_SERVER"
_RSE_JAVAOPTS="$RSE_JAVAOPTS -Dldap.server.port=$_RSE_LDAP_PORT"
_RSE_JAVAOPTS="$RSE_JAVAOPTS -Dldap.ptc.group.name.suffix=$_RSE_PTC"
_RSE_JAVAOPTS="$RSE_JAVAOPTS -DISPF_OPTS='$_RSE_ISPF_OPTS'"
_RSE_JAVAOPTS="$RSE_JAVAOPTS -DA_PLUGIN_PATH=$RSE_LIB"
_RSE_JAVAOPTS="$RSE_JAVAOPTS -Xbootclasspath/p:$RSE_LIB/bidiTools.jar"
_RSE_JAVAOPTS="$RSE_JAVAOPTS -Dfile.encoding=$_RSE_HOST_CODEPAGE"
_RSE_JAVAOPTS="$RSE_JAVAOPTS -Dconsole.encoding=$_RSE_HOST_CODEPAGE"

_RSE_JAVAOPTS="$RSE_JAVAOPTS -DDSTORE_INITIAL_SIZE=0"
_RSE_JAVAOPTS="$RSE_JAVAOPTS -DDSTORE_MAX_FREE=0"
_RSE_JAVAOPTS="$RSE_JAVAOPTS -DDSTORE_SPIRIT_ON=true"
_RSE_JAVAOPTS="$RSE_JAVAOPTS -DSPIRIT_EXPIRY_TIME=90"
_RSE_JAVAOPTS="$RSE_JAVAOPTS -DSPIRIT_INTERVAL_TIME=6"
_RSE_JAVAOPTS="$RSE_JAVAOPTS -Dcom.ibm.cacheLocalHost=true"
_RSE_JAVAOPTS="$RSE_JAVAOPTS -Duser.home=$HOME"
_RSE_JAVAOPTS="$RSE_JAVAOPTS -Dclient.username=$RSE_USER_ID"
_RSE_JAVAOPTS="$RSE_JAVAOPTS -Dlow.heap.usage.ratio=15"
_RSE_JAVAOPTS="$RSE_JAVAOPTS -Dmaximum.heap.usage.ratio=40"
_RSE_JAVAOPTS="$RSE_JAVAOPTS -DDSTORE_KEEPALIVE_ENABLED=true"
_RSE_JAVAOPTS="$RSE_JAVAOPTS -DDSTORE_KEEPALIVE_RESPONSE_TIMEOUT=60000"
_RSE_JAVAOPTS="$RSE_JAVAOPTS -DDSTORE_IO_SOCKET_READ_TIMEOUT=180000"
_RSE_JAVAOPTS="$RSE_JAVAOPTS -DRSECOMM_LOGFILE_MAX=0"
_RSE_JAVAOPTS="$RSE_JAVAOPTS -Djob.monitor.port=$_RSE_JMON_PORT"
_RSE_JAVAOPTS="$RSE_JAVAOPTS -Dlock.info.timeout=10000"
_RSE_JAVAOPTS="$RSE_JAVAOPTS -showversion"
_RSE_SERVER_CLASS=org.eclipse.dstore.core.server.Server
_RSE_DAEMON_CLASS=com.ibm.etools.zos.server.RseDaemon
_RSE_POOL_SERVER_CLASS=com.ibm.etools.zos.server.ThreadPoolProcess
_RSE_SERVER_TIMEOUT=120000
_SCLMDT_BASE_HOME=$RSE_HOME
_SCLMDT_WORK_HOME=$CGI_ISPHOME
CGI_DTWORK=$_SCLMDT_WORK_HOME
_CMDSERV_BASE_HOME=$CGI_ISPHOME
_CMDSERV_CONF_HOME=$CGI_ISPCONF
_CMDSERV_WORK_HOME=$CGI_ISPWORK
#####
# (6) additional environment variables

```

図 13. *rsed.envvars*: RSE 構成ファイル (続き)

注: *rsed.envvars* で値およびディレクトリを指定する際に、シンボルが *rsed.envvars* に定義される限り、シンボリック・リンクを使用できます。

以下の定義が必要です。

JAVA_HOME

Java ホーム・ディレクトリ。デフォルトは `/usr/lpp/java/J6.0` です。
使用する Java インストール済み環境に合わせて変更してください。

RSE_HOME

RSE ホーム・ディレクトリー。デフォルトは /usr/lpp/rdz です。使用する Developer for System z インストール済み環境に合わせて変更してください。

_RSE_RSED_PORT

RSE デーモン・ポート番号。デフォルトは 4035 です。必要であれば変更できます。

注:

- ポートを選択する前に、ご使用のシステムで、そのポートが TSO コマンド NETSTAT および NETSTAT PORTL で使用可能であることを確認してください。
- このポートは、クライアント/ホスト通信に使用されます。
- RSED 開始タスクは、ここで指定されたポート番号をオーバーライドできます。

_RSE_JMON_PORT

JES ジョブ・モニター・ポート番号。デフォルトは 6715 です。必要であれば変更できます。

注:

- この値は、FEJCNFG 構成ファイル内で JES ジョブ・モニターに設定したポート番号に一致する必要があります。これらの値が異なる場合、RSE はクライアントを JES ジョブ・モニターに接続できません。JES ジョブ・モニターに対して変数を定義する方法については、38 ページの『FEJCNFG、JES ジョブ・モニター構成ファイル』を参照してください。
- ポートを選択する前に、ご使用のシステムで、そのポートが TSO コマンド NETSTAT および NETSTAT PORTL で使用可能であることを確認してください。
- このポート上のすべての通信は、ご使用の z/OS ホスト・システムだけに限定されます。

RSE_HLQ

Developer for System z をインストールするために使用する高位修飾子。デフォルトは FEK です。使用する Developer for System z データ・セットのロケーションに合わせて変更してください。

_RSE_HOST_CODEPAGE

ホスト・システムのコード・ページ。デフォルトは IBM-1047 です。使用するホスト・システム・コード・ページに合わせて変更してください。このコード・ページはデータ変換処理には使用されず、サーバーの操作およびクライアント接続のセットアップにのみ使用されることに注意してください。データ変換処理に使用するコード・ページは、Developer for System z クライアントから指定されます (これは、「MVS ファイル」サブシステムのプロパティから取得されます)。

TZ

タイム・ゾーン・セレクター。デフォルトは EST5EDT です。デフォルトのタイム・ゾーンは、UTC -5 時間 (東部標準時 (EST) 東部夏時間 (EDT)) です。使用するタイム・ゾーンに合わせて変更してください。

詳細は、「UNIX System Services コマンド解説書」(SA88-8641)に記載されています。

LANG デフォルト・ロケールの名前を指定します。デフォルトは C です。C は POSIX ロケールを指定し、(例えば) Ja_JP は日本語ロケールを指定します。使用するロケールに合わせて変更してください。

PATH コマンド・パス。デフォルトは /bin:/usr/sbin:. です。必要であれば変更できます。

_CEE_DMPTARG

Java 仮想マシン (JVM) によって使用される、言語環境プログラム (LE) z/OS UNIX ダンプ・ロケーション。デフォルトは /tmp です。

STEPLIB

LINKLIST/LPALIB でなく、MVS データ・セットにアクセスします。デフォルトは NONE です。

以下の 1 つ以上の STEPLIB ディレクティブのコメントを外してカスタマイズすることにより、前提条件のライブラリーを LINKLIST/LPALIB 内に保持せずに済ませることができます。以下にリストしたライブラリーの詳しい使用方法については、21 ページの『PARMLIB の変更』を参照してください。

```
# RSE
STEPLIB=$STEPLIB:CEE.SCEERUN:CEE.SCEERUN2:CBC.SCLBDLL
# ISPF
STEPLIB=$STEPLIB:ISP.SISPLOAD:ISP.SISPLPA:SYS1.LINKLIB
# SCLM Developer Toolkit
STEPLIB=$STEPLIB:$RSE_HLQ.SFEKAUTH:$RSE_HLQ.SFEKLOAD
# zUnit, xUnit support for Enterprise COBOL and PL/I
STEPLIB=$STEPLIB:$RSE_HLQ.SFEKLOAD:SYS1.CSSLIB:SYS1.SIXMLOD1
```

注:

- STEPLIB を z/OS UNIX で使用すると、パフォーマンスに悪い影響が出ます。
- 1 つの STEPLIB ライブラリーに APF 許可を与える場合、他のすべての STEPLIB ライブラリーにも APF 許可を与える必要があります。ライブラリーは、STEPLIB 内で許可のないライブラリーと混用した場合、APF 許可を失います。
- LPA 配置用に設計されたライブラリーは、LINKLIST または STEPLIB によってアクセスされる場合に、追加のプログラム制御および APF 許可を必要とすることがあります。
- サーバー JCL 内の STEPLIB DD ステートメントのコーディングでは、要求された STEPLIB 連結は設定されません。

_RSE_JAVAOPTS

追加の RSE 固有の Java オプション。この定義について詳しくは、56 ページの『_RSE_JAVAOPTS での追加 Java 始動パラメーターの定義』を参照してください。

TSO コマンド・サービスまたは SCLM Developer Toolkit に ISPF の TSO/ISPF クライアント・ゲートウェイが使用される場合は、以下の定義が必要です。

CGI_ISPHOME

TSO/ISPF クライアント・ゲートウェイ・サービスを提供する ISPF コードのホーム・ディレクトリー。デフォルトは `/usr/lpp/ispf` です。使用する ISPF インストール済み環境に合わせて変更してください。このディレクティブは、ISPF の TSO/ISPF クライアント・ゲートウェイを使用する場合にのみ必要です。

CGI_ISPCONF

ISPF 基本構成ディレクトリー。デフォルトは `/etc/rdz` です。TSO/ISPF クライアント・ゲートウェイ・カスタマイズ・ファイル、`ISPF.conf` のロケーションに合わせて変更してください。このディレクティブは、ISPF の TSO/ISPF クライアント・ゲートウェイを使用する場合にのみ必要です。

CGI_ISPWORK

ISPF 基本作業ディレクトリー。デフォルトは `/var/rdz` です。TSO/ISPF クライアント・ゲートウェイが使用する `WORKAREA` ディレクトリーのロケーションに合わせて変更してください。このディレクティブは、ISPF の TSO/ISPF クライアント・ゲートウェイを使用する場合にのみ必要です。

注:

- CGI_ISPWORK で指定したパスには、TSO/ISPF クライアント・ゲートウェイによって `/WORKAREA` が追加されます。手動では追加しないでください。
- カスタマイズ可能な環境の構築に `SFEKSAMP(FEKSETUP)` サンプル・ジョブを使用しなかった場合は、CGI_ISPWORK で指定したパスに `WORKAREA` ディレクトリーが存在することを確認してください。ディレクトリー許可ビットは 777 でなければなりません。

STEPLIB

STEPLIB については、以前に、必要な定義のセクションで説明しました。

_RSE_ISPF_OPTS

追加の TSO/ISPF クライアント・ゲートウェイ固有の Java オプション。デフォルトは `"` です。この定義について詳しくは、66 ページの

『_RSE_ISPF_OPTS での追加 Java 始動パラメーターの定義』を参照してください。このディレクティブは、ISPF の TSO/ISPF クライアント・ゲートウェイを使用する場合にのみ必要です。

CGI_ISPPREF

TSO/ISPF クライアント・ゲートウェイによって作成された一時データ・セットの高位修飾子。デフォルトは `"&SYSPREF..ISPF.VCMISPF"` です。コメント解除し、ご使用のデータ・セットの命名規則に合わせて変更してください。このディレクティブは、ISPF の TSO/ISPF クライアント・ゲートウェイを使用する場合にのみ必要です。

データ・セット名の中で、以下の変数を使用できます。

- `&SYSUID`。開発者のユーザー ID の代わりに使用します。
- `&SYSPREF`。開発者の TSO 接頭部、または (TSO 接頭部を判別できない場合には) ユーザー ID の代わりに使用します。
- `&SYSNAME`。IEASYMxx parmlib メンバーで指定されたシステム名の代わりに使用します。

注: このディレクティブには ISPF APAR OA38740 が必要です。

SCLM Developer Toolkit を使用する場合は、以下の定義が必要です。

_SCLMDT_CONF_HOME

SCLM Developer Toolkit 基本構成ディレクトリー。デフォルトは /var/rdz/sclmdt です。SCLMDT が SCLM プロジェクト情報を保管するために使用する CONFIG ディレクトリーのロケーションに合わせて変更してください。このディレクティブは、SCLMDT を使用する場合にのみ必要です。

注: SCLMDT_CONF_HOME に指定したパスには、SCLMDT によって /CONFIG と /CONFIG/PROJECT が追加されます。手動では追加しないでください。

STEPLIB

STEPLIB については、以前に、必要な定義のセクションで説明しました。

_SCLMDT_TRANTABLE

ロング/ショート・ネーム変換 VSAM の名前。デフォルトは FEK.#CUST.LSTRANS.FILE です。コメントを外し、ISP.SISPSAMP (FLM02LST) SCLM サンプル・ジョブで使用される名前に合わせて変更してください。このディレクティブが必要になるのは、SCLM Developer Toolkit でロング/ショート・ネーム変換を使用する場合だけです。

ANT_HOME

Ant インストールのホーム・ディレクトリー。デフォルトは /usr/lpp/apache/Ant/apache-ant-1.7.1 です。使用する Ant インストールに合わせて変更してください。このディレクティブが必要になるのは、SCLM Developer Toolkit で Java EE ビルド・サポートを使用する場合だけです。

以下の定義はオプションです。省略した場合は、デフォルト値が使用されます。

_RSE_PORTRANGE

RSE サーバーがクライアントとの通信用に開くことができるポート範囲を指定します。デフォルトでは、任意のポートを使用できます。この定義について詳しくは、55 ページの『RSE サーバーに使用可能な PORTRANGE の定義』を参照してください。これは、オプションのディレクティブです。

_BPXK_SETIBMOPT_TRANSPORT

使用する TCP/IP スタックの名前を指定します。デフォルトは TCPIP です。コメント解除して、要求される TCP/IP スタック名に変更してください。この名前は、関連する TCPIP.DATA 内の TCPIPJOBNAME ステートメントで定義されています。これは、オプションのディレクティブです。

注:

- サーバー JCL 内の SYSTCPD DD ステートメントのコーディングでは、要求されたスタックのアフィニティーは設定されません。
- このディレクティブがアクティブでないときは、RSE がシステム上の使用可能なすべてのスタックにバインドします (BIND INADDRANY)。

TMPDIR

一時ファイルの保管に使用するパスを指定します。デフォルトは /tmp で

す。要求されたパスを使用するには、コメント解除して変更します。これは、オプションのディレクティブです。

_RSE_FEK_SAF_CLASS

FEK.* プロファイルが定義されているセキュリティー・クラスを指定します。デフォルトは FACILITY です。指定した値を使用させるには、コメントを外して変更します。これは、オプションのディレクティブです。

_RSE_LDAP_SERVER

クライアントへのプッシュ機能で使用する LDAP サーバー・ホスト名を指定します。デフォルトは、現在の z/OS ホスト名です。指定した値を使用させるには、コメントを外して変更します。これは、オプションのディレクティブです。

_RSE_LDAP_PORT

クライアントへのプッシュ機能で使用する LDAP サーバー・ポートを指定します。デフォルトは 389 です。指定した値を使用させるには、コメントを外して変更します。これは、オプションのディレクティブです。

_RSE_LDAP_PTC_GROUP_SUFFIX

LDAP サーバー内でクライアントへのプッシュ・グループを見つけるために必要なサフィックス「O=<organization>,C=<country>」を指定します。デフォルトは O=PTC,C=DeveloperForZ です。指定した値を使用させるには、コメントを外して変更します。これは、オプションのディレクティブです。

GSK_CRL_SECURITY_LEVEL

証明書の妥当性検査で、取り消された証明書がないかどうか CRL を検査するために、SSL アプリケーションが LDAP サーバーとの接続時に使用するセキュリティーのレベルを指定します。デフォルトは MEDIUM です。指定した値を使用させるには、コメントを外して変更します。これは、オプションのディレクティブです。以下の値が有効です。

- LOW: LDAP サーバーに接続できない場合でも、証明書の妥当性検査は失敗しません。
- MEDIUM: 証明書の妥当性検査を行うには LDAP サーバーが接続可能であることが必要ですが、CRL を定義する必要はありません。この値はデフォルトです。
- HIGH: 証明書の妥当性検査を行うには LDAP サーバーが接続可能であり、CRL を定義する必要があります。

注: このディレクティブには z/OS 1.9 以降が必要です。

GSK_LDAP_SERVER

1 つ以上の LDAP サーバー・ホスト名をブランクで区切って指定します。指定した LDAP サーバーを使用させて、それらのサーバーの CRL を取得するには、コメントを外して変更します。これは、オプションのディレクティブです。

ホスト名は TCP/IP アドレスと URL のどちらでもかまいません。それぞれのホスト名は、コロン (:) でホスト名と区切った、オプションのポート番号を含むことができます。

GSK_LDAP_PORT

LDAP サーバー・ポートを指定します。デフォルトは 389 です。指定した値を使用させるには、コメントを外して変更します。これは、オプションのディレクティブです。

GSK_LDAP_USER

LDAP サーバーに接続するときに使用する識別名を指定します。指定した値を使用させるには、コメントを外して変更します。これは、オプションのディレクティブです。

GSK_LDAP_PASSWORD

LDAP サーバーに接続するときに使用するパスワードを指定します。指定した値を使用させるには、コメントを外して変更します。これは、オプションのディレクティブです。

RSE_UBLD_DD

TSO または ISPF コマンドを呼び出す Developer for System z クライアントから IBM Rational Team Concert™ ユーザー・ビルド用 JCL を生成する際に使われる DD ステートメントを指定します。デフォルトで、Developer for System z は ISPF.conf (rsed.envvars 内の CGI_ISPCONF によって参照される) の定義を使用します。指定されているファイルでコメントを外し、DD 定義を使用するように変更します。『*ISPF.conf*、*ISPF* の *TSO/ISPF* クライアント・ゲートウェイ構成ファイル』で指定されている構文規則に従う必要があります。これは、オプションのディレクティブです。

RSE_UBLD_STEPLIB

TSO または ISPF コマンドを呼び出す Developer for System z クライアントから IBM Rational Team Concert ユーザー・ビルド用 JCL を生成する際に使われる STEPLIB ステートメントを指定します。デフォルトでは、Developer for System z は rsed.envvars 内の STEPLIB 定義を使用します。コメントを外して、指定されている STEPLIB 定義を使用するように変更します。これは、オプションのディレクティブです。

以下の定義は必須であり、IBM サポートから指示された場合以外、変更しないでください。

_CEE_RUNOPTS

言語環境プログラム (LE) ランタイム・オプション。デフォルトは "ALL31(ON) HEAP(32M,32K,ANYWHERE,KEEP,,) TRAP(ON)" です。変更しないでください。

_BPX_SHAREAS

シェルと同じアドレス・スペースでフォアグラウンド・プロセスを実行します。デフォルトは YES です。変更しないでください。

_BPX_SPAWN_SCRIPT

シェル・スクリプトを spawn() 関数から直接実行します。デフォルトは YES です。変更しないでください。

_EDC_ADD_ERRNO2

理由コードを z/OS UNIX エラー・メッセージに表示します。デフォルトは 1 です。変更しないでください。

JAVA_PROPAGATE

スレッド作成時にセキュリティーおよびワークロード・コンテキストを伝搬します (Java バージョン 1.4 以前専用)。デフォルトは NO です。変更しないでください。

RSE_DSN_SFEKLOAD

SFEKLOAD ロード・ライブラリーの完全修飾データ・セット名。デフォルトは \$RSE_HLQ.SFEKLOAD です。変更しないでください。

RSE_LIB

RSE ライブラリー・パス。デフォルトは \$RSE_HOME/lib です。変更しないでください。

PATH コマンド・パス。デフォルトは `.:$JAVA_HOME/bin:$RSE_HOME/`

`bin:$CGI_ISPHOME/bin:$PATH` です。変更しないでください。

LIBPATH

ライブラリー・パス。デフォルトは長すぎるので、ここには繰り返しません。変更しないでください。

CLASSPATH

クラス・パス。デフォルトは長すぎるので、ここには繰り返しません。変更しないでください。

_RSE_ISPF_OPTS

追加の TSO コマンド・サービス固有の Java オプション。デフォルトは "&SESSION=SPAWN\$_RSE_ISPF_OPTS" です。変更しないでください。

_RSE_JAVAOPTS

追加の RSE 固有の Java オプション。デフォルトは長すぎるので、ここには繰り返しません。変更しないでください。

_RSE_SERVER_CLASS

RSE サーバーの Java クラス。デフォルトは `org.eclipse.dstore.core.server.Server` です。変更しないでください。

_RSE_DAEMON_CLASS

RSE デーモンの Java クラス。デフォルトは `com.ibm.etools.zos.server.RseDaemon` です。変更しないでください。

_RSE_POOL_SERVER_CLASS

RSE スレッド・プールの Java クラス。デフォルトは `com.ibm.etools.zos.server.ThreadPoolProcess` です。変更しないでください。

_RSE_SERVER_TIMEOUT

RSE サーバーの (クライアントを待つ) タイムアウト値 (ミリ秒単位)。デフォルトは 120000 (2 分) です。変更しないでください。

SCLMDT_BASE_HOME

SCLM Developer Toolkit コードのホーム・ディレクトリー。デフォルトは \$RSE_HOME です。変更しないでください。

SCLMDT_WORK_HOME

SCLM Developer Toolkit 基本作業ディレクトリー。デフォルトは \$CGI_ISPHOME です。変更しないでください。

CGI_DTWORk

古いクライアント用の SCLM Developer Toolkit サポート。デフォルトは `$_SCLMDT_WORK_HOME` です。変更しないでください。

_CMDSERV_BASE_HOME

ISPF TSO/ISPF クライアント・ゲートウェイ・サービス・サポート。デフォルトは `$CGI_ISPHOME` です。変更しないでください。

_CMDSERV_CONF_HOME

ISPF TSO/ISPF クライアント・ゲートウェイ・サービス・サポート。デフォルトは `$CGI_ISPCONF` です。変更しないでください。

_CMDSERV_WORK_HOME

ISPF TSO/ISPF クライアント・ゲートウェイ・サービス・サポート。デフォルトは `$CGI_ISPWORK` です。変更しないでください。

RSE サーバーに使用可能な PORTRANGE の定義

これは、RSE サーバーがクライアントと通信できるポートを指定する、`rsed.envvars` カスタマイズの一環です。このポート範囲は、RSE デーモン・ポートとは関係ありません。

ポートの使用法を理解しやすいように、以下で RSE の接続プロセスを簡単に説明します。

1. クライアントは、ホスト・システム・ポート 4035、RSE デーモンに接続します。
2. RSE デーモンは、RSE サーバー・スレッドを作成します。
3. RSE サーバーは、クライアントが接続するホスト・システム・ポートを開きます。このポートの選択は、`rsed.envvars` の `_RSE_PORTRANGE` 定義を使用して構成できます。
4. RSE デーモンは、クライアントにポート番号を返します。
5. クライアントは、ホスト・システム・ポートに接続します。

注:

- このプロセスは、REXEC/SSH を使用したオプションの代替接続方式とほとんど同じです。
- 詳しくは、「ホスト構成リファレンス」(SA88-4226) の『Developer for System z について』を参照してください。

クライアントが z/OS と通信できるように、ポート範囲を指定するには、`rsed.envvars` 内で次の行をコメント解除し、カスタマイズします。

```
#_RSE_PORTRANGE=8108-8118
```

注: ポート範囲を選択する前に、**NETSTAT** および **NETSTAT PORTL** コマンドで、その範囲がシステム上で使用可能であることを確認してください。

PORTRANGE のフォーマットは `_RSE_PORTRANGE=min-max` です。max の値は範囲に含まれません。例えば、`_RSE_PORTRANGE=8108-8118` という式は、8108 から 8117 までのポート番号が使用可能であることを意味します。RSE サーバーによって使用されるポート番号は、次の順序で決定されます。

1. ゼロ以外のポート番号がクライアント上のサブシステム・プロパティーで指定されている場合は、指定したポート番号が使用されます。ポートが使用不可であると、接続は失敗します。このセットアップは、お勧めできません。

注: ホスト・システムでは、`rsed.envvars` に `deny.nonzero.port=true` ディレクティブを指定することで、このタイプの接続要求を拒否できます。このディレクティブについて詳しくは、『`_RSE_JAVAOPTS` での追加 Java 始動パラメーターの定義』を参照してください。

2. サブシステム・プロパティー内のポート番号が 0 の場合、しかも `_RSE_PORTRANGE` が `rsed.envvars` で指定されている場合は、`_RSE_PORTRANGE` で指定されたポート範囲が使用されます。範囲内で使用可能なポートがない場合、接続は失敗します。

RSE サーバーは、クライアントが接続している間中、ポートを独占的に使用するわけではありません。他の RSE サーバーがそのポートにバインドできないのは、サーバーのバインドからクライアントの接続までの間だけです。つまり、ほとんどの接続が範囲内の最初のポートを使用し、同じ範囲内の残りのポートは、同時に複数のログオンが発生したときのバッファーになる、ということです。

3. サブシステム・プロパティー内のポート番号が 0 で、`_RSE_PORTRANGE` が `rsed.envvars` の中で指定されていない場合は、使用可能な任意のポートが使用されます。

`_RSE_JAVAOPTS` での追加 Java 始動パラメーターの定義

`rsed.envvars` は、さまざまな `_RSE_*OPTS` ディレクティブにより、RSE プロセスの始動時に Java に追加パラメーターを指定することを可能にしています。`rsed.envvars` に含まれているサンプル・オプションは、それらをコメント解除することによってアクティブにできます。

`_RSE_JAVAOPTS` は、標準および RSE 固有の Java オプションを定義します。

`_RSE_JAVAOPTS=""`

変数初期化。変更しないでください。

`_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Xms128m -Xmx512m"`

初期 (Xms) および最大 (Xmx) ヒープ・サイズを設定します。デフォルトはそれぞれ、128M と 512M です。必要なヒープ・サイズ値が使用されるように変更してください。このディレクティブがコメント化されている場合は、Java のデフォルト値が使用されます。デフォルト値は、4M と 512M です (Java 5.0 の場合は 1M と 64M)。

注: このディレクティブの最適値を決定するには、「ホスト構成リファレンス」(SA88-4226) の『主要なリソース定義』を参照してください。

`_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Ddaemon.log=/var/rdz/logs"`

RSE デーモンおよびサーバーのログ・ファイル、および RSE 監査データに至るディレクトリー。デフォルトは `/var/rdz/logs` です。必要なロケーションが使用されるように変更してください。このディレクティブがコメント化されている場合は、RSE デーモンに割り当てたユーザー ID のホーム・ディレクトリーが使用されます。ホーム・ディレクトリーはユーザー ID の OMVS セキュリティー・セグメントで定義されます。

注:

- このディレクティブ、または対応するホーム・ディレクトリーに絶対パスが指定されていない (パスがスラッシュ (/) で始まっていない) 場合、実際のログ・ロケーションは、構成ディレクトリー (デフォルトでは /etc/rdz) に対する相対位置となります。
- サーバー・ログの完全なパスは daemonlog/server (ここで、daemonlog は daemon.log ディレクティブの値) です。

`_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Duser.log=/var/rdz/logs"`

ユーザー固有のログがあるディレクトリー。デフォルトは /var/rdz/logs です。必要なロケーションが使用されるように変更してください。このディレクティブがコメント化されている、または値がヌル・ストリングである場合は、クライアント・ユーザー ID のホーム・ディレクトリーが使用されます。ホーム・ディレクトリーはユーザー ID の OMVS セキュリティー・セグメントで定義されます。

注:

- このディレクティブ、または対応するホーム・ディレクトリーに絶対パスが指定されていない (パスがスラッシュ (/) で始まっていない) 場合、実際のログ・ロケーションは、構成ディレクトリー (デフォルトでは /etc/rdz) に対する相対位置となります。
- ユーザー・ログの完全パスは userlog/dstorelog/\$LOGNAME/ です。ここで、userlog は user.log ディレクティブの値、dstorelog は DSTORE_LOG_DIRECTORY ディレクティブの値、\$LOGNAME は大文字で表記されたクライアントのユーザー ID です。
- 各クライアントが \$LOGNAME を作成できるように、userlog/dstorelog の許可ビットが設定されていることを確認してください。

`_RSE_JAVAOPTS="$_RSE_JAVAOPTS -DDSTORE_LOG_DIRECTORY=`

このディレクトリーは、user.log ディレクティブに指定されているパスに付加されます。ディレクトリーが付加されたパスは、ユーザー固有のログへのパスとなります。デフォルトはヌル・ストリングです。指定したディレクトリーを強制的に使用させるには、変更してください。このディレクティブがコメント化されている場合は、.eclipse/RSE/ が使用されます。

注:

- ユーザー・ログの完全パスは userlog/dstorelog/\$LOGNAME/ です。ここで、userlog は user.log ディレクティブの値、dstorelog は DSTORE_LOG_DIRECTORY ディレクティブの値、\$LOGNAME は大文字で表記されたクライアントのユーザー ID です。
- ここで指定するディレクトリーは、user.log に指定したディレクトリーを基準とする相対位置であるため、通常、先頭はスラッシュ (/) ではありません。
- 各クライアントが \$LOGNAME を作成できるように、userlog/dstorelog の許可ビットが設定されていることを確認してください。

`_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Dlog.retention.period=5"`

デーモン・ログおよびユーザー・ログが保持される日数。デフォルトは 5 です。所定の日数を過ぎた後にログを削除するには、このディレクティブを

カスタマイズします。制限を設定しない場合は 0 を指定します。最大値は 365 です。デーモン・アクティビティを必要とするアクションが次に実行されるとき、デーモン・ログのクリーンアップが発生することに注意してください。ユーザー・ログは、次にユーザーが接続したときにクリーンアップされます。

以下のディレクティブは、デフォルトではコメント化されています。

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Dmaximum.clients=30"

1 つのスレッド・プールでサービスできるクライアントの最大数。デフォルトは 30 です。1 スレッド・プール当たりのクライアント数を制限するには、コメントを外してカスタマイズします。他の制限のために、RSE がこの限度に到達しない場合もあります。

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Dmaximum.threads=520"

新規クライアントを許可するための、1 つのスレッド・プールでのアクティブ・スレッド数の最大値。デフォルトは 520 です。1 スレッド・プール当たりのクライアント数を、使用中のスレッド数に基づいて制限するには、コメントを外してカスタマイズします。それぞれのクライアント接続では複数 (17 以上) のスレッドが使用されます。また、他の制限のために RSE がこの限度に到達しない場合があります。

注: この値は、SYS1.PARMLIB(BPXPRMxx) での MAXTHREADS および MAXTHREADTASKS の設定より小さくする必要があります。

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Dminimum.threadpool.process=1"

アクティブ・スレッド・プールの最小数。デフォルトは 1 です。少なくとも指定した数のスレッド・プール・プロセスを開始するには、コメントを外してカスタマイズします。スレッド・プール・プロセスは、RSE サーバー・スレッドのロード・バランシングに使用されます。より多くの新規プロセスが必要になった場合は、その時点で新規プロセスが開始されます。前もって新規プロセスを開始しておく、接続遅延を回避できますが、アイドル時間中に使用されるリソースが増えます。

注: single.logon ディレクティブがアクティブな場合、minimum.threadpool.process が 1 に設定されているとしても、少なくとも 2 つのスレッド・プールが開始されます。rsed.envvars 内の single.logon のデフォルト設定はアクティブです。

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Dmaximum.threadpool.process=100"

アクティブ・スレッド・プールの最大数。デフォルトは 100 です。スレッド・プール・プロセスの数を制限するには、コメントを外してカスタマイズします。スレッド・プール・プロセスは、RSE サーバー・スレッドのロード・バランシングに使用されます。このため、スレッドを制限すると、アクティブなクライアント接続の数も制限されます。

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Dipv6=true"

TCP/IP バージョン。デフォルトは false で、IPv4 インターフェースが使用されることを意味します。IPv6 インターフェースを使用するには、コメントを外して true を指定します。

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Denable.dDVIPA=true"

Distributed Dynamic VIPA サポート。デフォルトは false です。TCP/IP

の Distributed Dynamic VIPA により複数システム上のサーバーを単一のサーバーのように見せることができます。これを使用するには、コメントを外して `true` を指定します。

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Ddisplay.users=true"

アクティブ・ユーザーの自動表示。デフォルトは `false` です。各ユーザーのログオンおよびログオフのたびに `rserver.log` へのアクティブ・ユーザーの自動表示を有効にするには、コメントを外して `true` を指定します。

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Dkeep.all.logs=false"

デーモン・ログとユーザー・ログでタイム・スタンプを埋め込んだファイル名を使用します。デフォルトは `true` で、これは `log.retention.period` 設定によって削除されるまでログが保持されることを意味します。固定のログ・ファイル名を使用するには、コメントを外して `false` を指定します。このログ・ファイル名は、デーモンが開始するかユーザーが接続するたびに置き換えられます。

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Dkeep.last.log=true"

前のセッションに属するホスト・ログ・ファイルのコピーを保持します。デフォルトは `false` です。サーバーを始動してクライアントに接続するときに、前のログ・ファイルの名前を `*.last` に変更する場合は、コメントを外して `true` を指定します。`.dstore*` ユーザー・トレース・ファイルは、クライアントの再接続では自動的に削除されないことにご注意ください。また、`keep.last.log` 処理の一部でもありません。これらのファイルの除去は手動で行います。`keep.last.log` を有効にするには、`keep.all.logs` ディレクティブを `false` に設定する必要があります。

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Denable.standard.log=true"

スレッド・プールの `STDOUT` および `STDERR` ストリームをログ・ファイルに書き込みます。デフォルトは `false` です。`STDOUT` および `STDERR` ストリームを保存する場合は、コメントを外して `true` を指定します。結果のログ・ファイルは、`daemon.log` ディレクティブで参照されるディレクトリーに置かれます。

注: **MODIFY RSESTANDARDLOG** オペレーター・コマンドを使用して、ストリーム・ログ・ファイルの更新を動的に開始または停止することができます。

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Dlog.file.mode=RW.N.N"

ログ・ファイルおよびログ・ディレクトリーのアクセス許可マスク。デフォルトは `RW.N.N` で、所有者に読み取り/書き込みアクセス権を許可します。所有者のデフォルト・グループおよび他のすべてのユーザーにはアクセス権がありません。必要なアクセス権を設定するには、コメントを外してカスタマイズします。

UNIX 標準では、所有者、グループ、およびその他という 3 つのタイプのユーザーに対して権限を設定できることが定められています。この変数のフィールドは、この順序に一致しており、ピリオド (.) で区切られます。各フィールドは、空 (N と等しい) にするか、あるいは N、R、W、または RW を値として設定できます (N = なし、R = 読み取り、W = 書き込み)。

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Dlog.secure.mode=false"

ログ・ディレクトリーの所有権を検証します。デフォルトは `true` です。これにより、RSE はユーザー (RSE デーモン自体またはクライアントのユー

ザー ID) がログが書き込まれるディレクトリーの所有者であることを検証します。コメントを外し、`false` を指定して、このチェックをスキップし、ディレクトリーの所有権をチェックせずにログ・ファイルを書き込みます。テストが失敗した場合は、コンソール・メッセージ FEK301E が発行されます。

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Denable.port.of.entry=true"

Port Of Entry (POE) 検査オプション。デフォルトは `false` です。クライアント接続の POE 検査を施行するには、コメントを外して `true` を指定します。POE 検査のとき、クライアントの IP アドレスは、ご使用のセキュリティー・ソフトウェアによってネットワーク・アクセス・セキュリティー・ゾーンにマップされます。クライアント・ユーザー ID は、セキュリティー・ゾーンを定義しているプロファイルを使用する権限を持っている必要があります。

注:

- POE 検査は、使用しているセキュリティー製品でも有効にする必要があります。
- POE 検査を有効にすると、INETD など、他の z/OS UNIX サービスにも作用します。

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Denable.certificate.mapping=false"

ご使用のセキュリティー・ソフトウェアを使用して、X.509 証明書でログインを認証します。デフォルトは `true` です。セキュリティー・ソフトウェアの X.509 サポートに依存せず、RSE デーモンに認証を行わせるには、コメントを外して `false` を指定します。

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Denable.automount=true"

z/OS UNIX 自動マウントで作成されたホーム・ディレクトリーをサポートします。デフォルトは `false` です。z/OS UNIX 自動マウントでクライアント・ユーザー ID がディレクトリーの所有者として使用されるようにするには、コメントを外して `true` を指定します。

注: z/OS UNIX 自動マウントでは、ファイル・システムを作成する際にサービスを呼び出したプロセスのユーザー ID が使用されます。このオプションが使用不可になっている場合、そのプロセスは RSE スレッド・プール・サーバー (ユーザー ID は STCRSE) になります。このオプションを使用可能になっている場合は、サービスを起動する前に、クライアント・ユーザー ID を使用して新しい一時プロセスが作成されます。

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Denable.audit.log=true"

監査オプション。デフォルトは `false` です。クライアントが実行したアクションの監査ロギングを施行するには、コメントを外して `true` を指定します。監査ログは、RSE デーモンのログ・ロケーションに書き込まれます。そのロケーションを確認するには、`_RSE_JAVAOPTS` 変数の `daemon.log` オプションを参照してください。

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Daudit.cycle=30"

1 つの監査ログ・ファイルに保管される日数。デフォルトは 30 です。1 つの監査ログ・ファイルに書き込む監査データの量を制御するには、コメントを外してカスタマイズします。最大値は 365 です。

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Daudit.retention.period=0"

監査ログが保持される日数。デフォルトは 0 です。つまり、制限はありません。一定の日数を経た後に監査ログを削除するには、コメントを外してカスタマイズします。最大値は 365 です。

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Daudit.log.mode=RW.R.N"

監査ログのアクセス許可マスク。デフォルトは RW.R.N で、所有者に読み取り/書き込みアクセス権を許可します。所有者のデフォルト・グループは読み取りアクセス権を持ち、他のすべてのユーザーはアクセス権を持ちません。必要なアクセス権を設定するには、コメントを外してカスタマイズします。

UNIX 標準では、所有者、グループ、およびその他という 3 つのタイプのユーザーに対して権限を設定できることが定められています。この変数マスクのフィールドは、この順序に一致しており、ピリオド (.) で区切られます。各フィールドは、空 (N と等しい) にするか、あるいは N、R、W、または RW を値として設定できます (N = なし、R = 読み取り、W = 書き込み)。

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Daudit.action=<user exit>"

監査ログ・ファイルが閉じている場合に呼び出されるユーザー出口の名前。デフォルト値はありませんが、サンプルの出口が /usr/lpp/rdz/samples/process_audit.rex に提供されています。監査ログのポストプロセッシングを有効にするには、コメントを外し、ユーザー出口プログラムの絶対パス名を指定します。

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Daudit.action.id=<userid>"

audit.action 変数で指定された出口を実行するために使用されるユーザー ID。デフォルトは、RSE デーモンに割り当てられたユーザー ID です。監査ポストプロセッシング出口を実行するために指定 ID を使用するには、コメントを外し、ユーザー ID を指定します。

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Dlogon.action=<user_exit>"

ユーザーがログオンするときに呼び出されるユーザー出口の名前。デフォルト値はありませんが、サンプルの出口が /usr/lpp/rdz/samples/process_logon.sh に提供されています。ログオンのポストプロセッシングを有効にするには、コメントを外し、ユーザー出口プログラムの絶対パス名を指定します。

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Dlogon.action.id=<userid>"

logon.action 変数で指定された出口を実行するために使用されるユーザー ID。デフォルトは、RSE デーモンに割り当てられたユーザー ID です。ログオンのポストプロセッシング出口を実行するために指定 ID を使用するには、コメントを外し、ユーザー ID を指定します。

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Ddeny.nonzero.port=true"

クライアントによる通信ポート番号の選択を禁止します。デフォルトは false です。クライアントによって RSE サーバーで接続に使用するホスト・システム・ポートが指定される接続を拒否するには、コメントを外して true を指定します。詳しくは、55 ページの『RSE サーバーに使用可能な PORTRANGE の定義』を参照してください。

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Dsingle.logon=false"

1 つのユーザー ID による複数回のログオンを禁止します。デフォルトは true です。1 つのユーザー ID が 1 つの RSE デーモンに複数回ログオンできるようにするには、コメントを外して false を指定します。

注:

- このディレクティブがアクティブでないか、または true に設定されていない場合は、2 回目にログオンが試行されると 1 回目のログオン試行がホスト・システムによって取り消されます。この取り消し操作と同時にコンソール・メッセージ FEK210I が出ます。
- single.logon ディレクティブがアクティブな場合、minimum.threadpool.process が 1 に設定されていても、少なくとも 2 つのスレッド・プールが開始されます。rsed.envvars の minimum.threadpool.process のデフォルト設定は、1 です。

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Dprocess.cleanup.interval=0"

リカバリー不能エラーの状態にある RSE スレッド・プールを自動的に除去します。デフォルトでは、エラー状態のスレッド・プールは自動的に除去されません。エラー状態の RSE スレッド・プール・サーバーを一定の時間間隔 (秒単位) で自動的に除去するには、コメントを外してカスタマイズします。0 を指定するとインターバル・タイマーは開始しませんが、新規クライアント・ログオン中または DISPLAY PROCESS コマンドで RSE デーモンが RSE スレッド・プールをチェックする際に、エラー状態の RSE スレッド・プール・サーバーが除去されます。

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Dreject.logon.threshold=1000000"

指定されたサイズより大きいファイルを開いているスレッド・プールは、そのファイルがロードされるまで、新しいログオン要求を受け入れません。デフォルトのファイル・サイズは 1000000 バイトです。ファイル・サイズを指定して、そのサイズより大きいファイルが開かれているときにはスレッド・プールがログオン要求を無視するようにするには、コメントを外してカスタマイズします。他のスレッド・プールでは引き続き、新しいログオン要求を受け入れることができます。

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Dinclude.c=/etc/rdz/include.conf"

この変数は、C コードでのコンテンツ・アシスト用の強制インクルードのリストが入っている完全修飾された z/OS UNIX ファイルを指します。強制インクルードは、ファイルまたはメンバーがプリプロセッサ・ディレクティブを使用してソース・コードに組み込まれたかどうかに関係なく、コンテンツ・アシスト操作の実行時に構文解析されるファイルまたはディレクトリー、データ・セット、またはデータ・セット・メンバーで構成されます。構成ファイルの名前を指定するには、コメントを外してカスタマイズします。

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Dinclude.cpp=/etc/rdz/include.conf"

この変数は、C++ コードでのコンテンツ・アシスト用の強制インクルードのリストが入っている完全修飾された z/OS UNIX ファイルを指します。強制インクルードは、ファイルまたはメンバーがプリプロセッサ・ディレクティブを使用してソース・コードに組み込まれたかどうかに関係なく、コンテンツ・アシスト操作の実行時に構文解析されるファイルまたはディレクト

リー、データ・セット、またはデータ・セット・メンバーで構成されます。構成ファイルの名前を指定するには、コメントを外してカスタマイズします。

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -DCPP_CLEANUP_INTERVAL=60000"

未使用の C/C++ ヘッダー・ファイルのクリーンアップ間隔 (ミリ秒単位)。デフォルトは 60000、つまり 1 分です。クリーンアップ間隔を変更するには、コメントを外してカスタマイズします。値 0 を指定すると、C/C++ ヘッダー・ファイルのキャッシュが行われなため、エディターでのリモート・コンテンツ・アシストのパフォーマンスが低下します。

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -DRIS_BUFFER=8"

リモート索引の作成時に使用されるバッファー・サイズ (M バイト単位)。デフォルトは 8 MB です。バッファー・サイズを変更するには、コメントを外してカスタマイズします。有効な値は 1 から 2000 まで (両端の値を含む) の整数です。バッファーを大きくすると索引作成時間が短縮されますが、スレッド・プールの Java ヒープがより多く使用されます。索引の作成が終了する前にバッファーがいっぱいになると、バッファーは索引に自動的にフラッシュされます。

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -DAPPLID=FEKAPPL"

RSE サーバー・アプリケーション ID。デフォルトは FEKAPPL です。必要とするアプリケーション ID を使用させるには、コメントを外してカスタマイズします。

注:

- 使用するセキュリティー・ソフトウェアに対してアプリケーション ID を定義する必要があります。そうしないと、クライアントはログオンできません。
- この値を変更した場合のセキュリティーへの影響については、「[ホスト構成リファレンス](#)」(SA88-4226) の『[PassTickets の使用](#)』を参照してください。
- この値は、FEJJCNFG 構成ファイル内で JES ジョブ・モニターに設定したアプリケーション ID に一致する必要があります。これらの値が異なる場合、RSE はクライアントを JES ジョブ・モニターに接続できません。JES ジョブ・モニターに対して変数を定義する方法については、38 ページの『[FEJJCNFG、JES ジョブ・モニター構成ファイル](#)』を参照してください。

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Dkeep.stats.copy.local=true"

ローカル・コピー中に ISPF 統計を保持します。デフォルトは false です。これは、create-date および changed-by のような ISPF 統計がターゲット・データ・セットまたはメンバーで更新されることを意味します。コメントを外し、true を指定して、ソースとターゲットが同じホスト・システムにあるところでのコピー中に、元の ISPF 統計を保持します。

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Ddebug.miner.autoreconnect=0"

デバッグ・マネージャーに自動再接続します。デフォルトは 0 です。これは、デバッグ・マネージャー・サーバーとの接続が確立されなかったか、切断されたときに、毎分デバッグ・マイナーがデバッグ・マネージャーに再接

続しようとすることを意味しています。コメントを外して、異なる値を指定して、デバッグ・マイナーがデバッグ・マネージャーに接続しようとする頻度を制限します。

表 12. デバッグ・マネージャーへの自動再接続

debug.miner.autoreconnect	再接続動作
-1	再接続しない
0 (デフォルト)	成功するまで毎分再接続しようとする
1-86400	指定された回数まで再接続しようとする。最大値は 86400 であり、24 時間です。

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Ddebug.miner.localhost=localhost"

localhost の TCP/IP 定義の代替。デバッグ・マイナーは、localhost の指定を使用してデバッグ・マネージャーに接続しようとします。localhost がローカル・ループバック・アドレス (IPv4 の場合 127.0.0.1、IPv6 の場合 ::1) に解決しない場合、これは失敗します。必要な場合は、コメントを外して、ローカル・ループバック・アドレスを指定します。

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Dsearch.server.limit.hits=0"

索引付けされていないファイルおよびテキストの検索におけるリソース使用量を制限します。デフォルトは 0 (制限なし) です。指定した数の結果が検出されたら検索を停止するには、このディレクティブのコメントを外してカスタマイズします。

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Dsearch.server.limit.scanned_objects=0"

索引付けされていないファイルおよびテキストの検索におけるリソース使用量を制限します。デフォルトは 0 (制限なし) です。指定した数のオブジェクト (データ・セットまたは PDS(E) メンバー) がスキャンされたら検索を停止するには、このディレクティブのコメントを外してカスタマイズします。

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Dsearch.server.limit.timeout=0"

索引付けされていないファイルおよびテキストの検索におけるリソース使用量を制限します。デフォルトは 0 (制限なし) です。指定した秒数が経過したら検索を停止するには、このディレクティブのコメントを外してカスタマイズします。

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Dsearch.server.limit.lines=0"

索引付けされていないファイルおよびテキストの検索におけるリソース使用量を制限します。デフォルトは 0 (制限なし) です。指定した数の行がスキャンされたら検索を停止するには、このディレクティブのコメントを外してカスタマイズします。

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Dsearch.server.limit.errcount=true"

索引付けされていないファイルおよびテキストの検索におけるリソース使用量を制限します。デフォルトは false (制限なし) です。ゼロ以外の言語環境 (LE) ERRCOUNT 値を超える前に検索を停止するには、コメントを外して true を指定します。

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -DDISABLE_TEXT_SEARCH=true"

索引付けされていないテキストの検索ができないようにします。デフォルト

は false です。ユーザーがホスト上で全文検索を開始できないようにするには、コメントを外して true を指定します。

**#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS
-DDISABLE_REMOTE_INDEX_SEARCH=true"**

クライアントのリモート索引検索メニュー項目を使用不可にします。デフォルトは false です。ユーザーがホスト・システム・データ・セットのリモート索引を作成できないようにするには、コメントを外して true と指定します。このオプションは、クライアントのバージョン 8.5.1 以降でのみ機能します。

**#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS
-DDISABLE_DELETE_IN_SUBPROJECT=true"**

z/OS サブプロジェクトのコンテキスト・メニューで「削除」メニュー項目を使用不可にします。デフォルトは false です。ユーザーが z/OS サブプロジェクトのコンテキスト・メニューで「削除」メニュー項目を使用できないようにするには、コメントを外して true と指定します。このオプションは、クライアントのバージョン 8.0.1 以降でのみ機能します。

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -DDENY_PASSWORD_SAVE=true"

パスワード保存オプション。デフォルトは false です。ユーザーがホスト・パスワードをクライアントに保存できないようにするには、コメントを外して true を指定します。以前に保存されていたパスワードは、除去されます。このオプションは、クライアントのバージョン 7.1 以降でのみ機能します。

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -DHIDE_ZOS_UNIX=true"

z/OS UNIX オプションを隠します。デフォルトは false です。クライアント側でユーザーが z/OS UNIX のエレメント (つまりディレクトリー構造とコマンド行) を表示できないようにするには、コメントを外して true を指定します。このオプションは、クライアントのバージョン 7.6 以降でのみ機能します。

**#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS
-DDSTORE_IDLE_SHUTDOWN_TIMEOUT=3600000"**

活動停止中のクライアントを切断します。デフォルトでは、活動停止中のクライアントは切断されません。指定したミリ秒数の間、活動停止中のクライアントを切断するには、コメントを外してカスタマイズします。3600000 ミリ秒は 1 時間に相当します。

**#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS
-DDSTORE_USE_THREADED_MINERS=false"**

各マイナーを別個のスレッドで実行します。デフォルトは true です。ユーザー固有のすべてのマイナーを単一スレッドから実行する場合は、コメントを外して false を指定します。1 ユーザーのすべてのマイナーを単一スレッドにグループ化するとスレッド・プール内のスレッド使用量が削減されますが、マルチタスキングのユーザーのコマンド処理に若干の遅延が発生する可能性があります。

**#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS
-DDSTORE_SSL_ALGORITHM=TLSv1.2"**

クライアントと暗号化通信を行う際に SSL ではなく TLS を使用します。デフォルトは SSL です。Developer for System z クライアント/ホスト通信

において TLS (v1.2) の使用を開始するには、コメントを外して TLSv1.2 を指定します。このオプションは、Java バージョン 7.0 以降およびクライアントのバージョン 9.0 以降でのみ機能します。デフォルトでは、バージョン 9.0 のクライアントも SSL を使用することにご注意ください。クライアントでは、TLS 暗号化通信を使用するには eclipse.ini で -DDSTORE_SSL_ALGORITHM=TLSv1.2 が指定されている必要があります。

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -DDSTORE_TCP_NO_DELAY=true"

TCP/IP DELAY ACK 関数を使用不可にします。デフォルトは false です。Developer for System z クライアント/ホスト通信において TCP/IP の遅延 ACK が行われなくするには、コメントを外して true を指定します。

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -DDSTORE_TRACING_ON=true"

dstore トレースを開始します。IBM サポートから指示された場合にのみ使用してください。結果の .dstoreTrace ログ・ファイルは、EBCDIC ではなく Unicode (ASCII) で作成されます。

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -DDSTORE_MEMLOGGING_ON=true"

dstore メモリ・トレースを開始します。IBM サポートから指示された場合にのみ使用してください。結果の .dstoreMemLogging ログ・ファイルは、EBCDIC ではなく Unicode (ASCII) で作成されます。

_RSE_ISPF_OPTS での追加 Java 始動パラメーターの定義

rsed.envvars は、さまざまな _RSE_*OPTS ディレクティブにより、RSE プロセスの始動時に Java に追加パラメーターを指定することを可能にしています。rsed.envvars に含まれているサンプル・オプションは、それらをコメント解除することによってアクティブにできます。

_RSE_ISPF_OPTS ディレクティブは RSE 固有の Java オプションであり、デフォルトでは ISPF の TSO/ISPF クライアント・ゲートウェイが Developer for System z によって使用されている場合にのみ有効です。

_RSE_ISPF_OPTS=""

変数初期化。変更しないでください。

_RSE_ISPF_OPTS="\$_RSE_ISPF_OPTS&ISPPROF=&SYSUID..ISPPROF"

ISPF の初期化に既存の ISPF プロファイルを使用します。指定した ISPF プロファイルを使用するには、データ・セット名のコメントを外して変更します。

データ・セット名の中で、以下の変数を使用できます。

- &SYSUID。開発者のユーザー ID の代わりに使用します。
- &SYSPREF。開発者の TSO 接頭部または TSO 接頭部を判別できない場合にはユーザー ID の代わりに使用します。
- &SYSNAME。IEASYMxx parmlib メンバーで指定されたシステム名の代わりに使用します。

ISPF.conf、ISPF の TSO/ISPF クライアント・ゲートウェイ構成ファイル

ISPF の TSO/ISPF クライアント・ゲートウェイは、ISPF.conf 内の定義を使用して、バッチ TSO および ISPF コマンドを実行するための有効な環境を作成します。Developer for System z は、その環境を使用して、一部の MVS ベースのサービスを実行します。それらのサービスには、TSO コマンド・サービス、SCLM Developer Toolkit が含まれます。

ISPF.conf は /etc/rdz/ に置かれます。ただし、ジョブ FEK.SFEKSAMP(FEKSETUP) をカスタマイズして実行依頼したときに、別のロケーションを指定した場合は除きます。詳しくは、20 ページの『カスタマイズのセットアップ』を参照してください。このファイルは、TSO **OEDIT** コマンドで編集できます。

定義は列 1 から始める必要があります。US コード・ページを使用する場合、コメント行はアスタリスク (*) で始まります。データ行には、ディレクティブとそれに割り当てられる値のみを入れることができます。同じ行にコメントを入れることはできません。行の継続はサポートされていません。データ・セット名を連結するときは、それらを同じ行に追加し、名前同士をコンマ (,) で分離します。

ISPF データ・セットの正しい名前を指定するだけでなく、次の例に示すように、TSO コマンド・サービス・データ・セット名 FEK.SFEKPROC を SYSPROC ステートメントまたは SYSEXEC ステートメントに追加します。

```
* REQUIRED:
sysproc=ISP.SISPLIB,FEK.SFEKPROC
ispmlib=ISP.SISPMENU
isptlib=ISP.SISPTENU
ispplib=ISP.SISPPENU
ispslib=ISP.SISPSLIB
ispllib=ISP.SISLOAD

* OPTIONAL:
*allocjob = ISP.SISPSAMP(ISPZISP2)
*ISPF_timeout = 900
```

図 14. ISPF.conf: ISPF 構成ファイル

注:

- TSO 環境をカスタマイズするために、独自の DD のようなステートメントとデータ・セット連結を追加し、TSO ログオン・プロシーチャーを模倣することができます。詳しくは、「ホスト構成リファレンス」(SA88-4226) の『TSO 環境のカスタマイズ』を参照してください。
- STEPLIB ディレクティブを定義することはできません。代わりに、rsed.envvars 内の STEPLIB ディレクティブを使用してください。
- TSO/ISPF クライアント・ゲートウェイは、**ISPSTART** などの ISPF コマンドをインターセプトするサード・パーティー製品を使用した場合、正しく機能しない可能性があります。その製品を Developer for System z に対して無効にする方法については、製品の資料を参照してください。その製品が特定の DD ステートメントを DUMMY に割り振ることを必要とする場合は、その DD ステートメントを nullfile に割り振ることによって、この動作を ISPF.conf でシミュレートできます。

次に例を示します。

ISPTRACE=nullfile

- allocjob ディレクティブを使用する場合は、前に ISPF.conf で行った DD 定義を元に戻さないように注意してください。
- SMFPRMxx parmlib メンバー内の JWT パラメーターの設定が ISPF.conf 内の ISPF_timeout 値よりも小さい場合は、モジュール ISPZTS0 のシステム異常終了 522 が予想されます。これによって、Developer for System z の操作が影響を受けることはありません。必要なときには TSO/ISPF クライアント・ゲートウェイが自動的に再始動されるからです。
- 変更は、すべての新規呼び出しについてアクティブになります。サーバーの再始動は必要ありません。

オプションのコンポーネント

このセクションは、さまざまなオプションのカスタマイズ・タスクを結合したものです。求めるサービスを構成するには、該当するセクションの説明に従ってください。

Developer for Developer for System z スタンドアロン・コンポーネントへのカスタマイズ:

- 71 ページの『第 3 章 (オプション) 共通アクセス・リポジトリ・マネージャー (CARMA)』
- 109 ページの『第 4 章 (オプション) SCLM Developer Toolkit』
- 117 ページの『第 5 章 (オプション) Application Deployment Manager (非推奨)』
- 125 ページの『第 6 章 (オプション) ホスト・ベースのコード分析』

Developer for System z 構成ファイルへのカスタマイズ

- 129 ページの『(オプション) pushtoclient.properties、ホスト・ベースのクライアント制御』
- 133 ページの『(オプション) ssl.properties、RSE SSL 暗号化』
- 135 ページの『(オプション) rsecomm.properties、RSE トレース』
- 137 ページの『(オプション) include.conf、C/C++ コンテンツ・アシスト用の強制インクルード』

他の製品への Developer for System z 関連のカスタマイズ:

- 138 ページの『(オプション) z/OS UNIX サブプロジェクト』
- 139 ページの『(オプション) インクルード・プリプロセッサのサポート』
- 140 ページの『(オプション) Enterprise COBOL および PL/I での xUnit サポート』
- 141 ページの『(オプション) エンタープライズ・サービス・ツール・サポート』
- 142 ページの『(オプション) CICS 双方向言語サポート』
- 143 ページの『(オプション) 生成コードの診断用 IRZ メッセージ』
- 144 ページの『(オプション) 統合デバッガー』
- 152 ページの『(オプション) 問題判別ツールのサポート』
- 152 ページの『(オプション) DB2 および IMS のデバッグのサポート』
- 153 ページの『(オプション) File Manager のサポート』

インストール検査

各種インストール検査プログラム (IVP) についての詳細は、155 ページの『第 8 章 インストール検査』に記載されています。これは、一部の IVP がオプション・コンポーネント用であるためです。

以下のシナリオで基本機能をテストできます。

1. JMON 開始タスクまたはユーザー・ジョブを開始します。DD STDOUT での開始情報が、次のメッセージで終わらなければなりません。

```
FEJ211I Server ready to accept connections.
```

ジョブが戻りコード 66 で終了する場合は、FEK.SFEKAUTH に APF 許可がありません。

2. RSED 開始タスクまたはユーザー・ジョブを、IVP=IVP パラメーターを指定して開始します。このパラメーターを指定すると、サーバーはいくつかのインストール検査テストを行った後に終了します。DD STDOUT を確認して、以下の IVP が成功したことを示すメッセージの有無を調べます。

- Java の始動
- JES ジョブ・モニター接続
- TCP/IP のセットアップ

3. RSED 開始タスクまたはユーザー・ジョブを、IVP パラメーターを指定しないで開始します。始動が成功すると、RSE デーモンは次のコンソール・メッセージを発行します。

```
FEK002I RseDaemon started. (port=4035)
```

4. 以下のオペレーター・コマンドを発行し、結果のコンソール・メッセージで、テストが正常に実行されたことを確認します。

```
F RSED,APPL=IVP PASSTICKET,userid  
F RSED,APPL=IVP DAEMON,userid  
F RSED,APPL=IVP ISPF,userid
```

userid は、有効な TSO ユーザー ID に置き換えてください。

第 3 章 (オプション) 共通アクセス・リポジトリ・マネージャ ー (CARMA)

共通アクセス・リポジトリ・マネージャ (CARMA) は、Repository Access Manager (RAM) 向けのサーバー・プラットフォームです。RAM は、z/OS ベースの Software Configuration Manager (SCM) 用のアプリケーション・プログラミング・インターフェース (API) です。SCM 機能を RAM に内包することにより、サポートされる SCM にクライアントがアクセスするときに、単一の API が使用できるようになります。

Developer for System z は事前に組み込まれた複数の RAM と、ユーザーが独自の RAM を作成するためのソース・コード・サンプルを提供しています。

ホスト・システム・ベースの SCM は、サービスにアクセスするために単一ユーザーのアドレス・スペースを必要とします。つまり、CARMA は、ユーザーごとに CARMA サーバーを 1 つ始動する必要があります。複数のユーザーをサポートする単一のサーバーを作成することはできません。

要件およびチェックリスト

このカスタマイズ・タスクを完了するには、セキュリティ管理者および TCP/IP 管理者の支援が必要になります。このタスクには、以下のリソースまたは特殊なカスタマイズ・タスクが必要です。

- (オプション) 内部通信用の TCP/IP ポート範囲
- (オプション) 開発者に CARMA VSAM ファイルの更新機能の使用を許可するセキュリティ規則
- (オプション) ユーザーに CRA* ジョブの実行依頼を許可するセキュリティ規則
- (オプション) LPA 更新

ご使用のサイトで CARMA の使用を開始するには、以下のタスクを実行してください。特に断りがない限り、すべてのタスクは必須です。

1. CARMA を始動する方式を選択し、どの RAM をアクティブにするかを選択します。事前構成のセットアップとして、RAM とサーバー始動方式のいくつかの組み合わせが可能です。詳細については、72 ページの『サーバー始動方式とアクティブ RAM の選択』を参照してください。
2. CARMA VSAM データ・セットを作成します。詳細については、94 ページの『CARMA VSAM データ・セット』と 95 ページの『CARMA Repository Access Manager (RAM)』を参照してください。
3. CARMA とインターフェースするための RSE 構成ファイルの初期カスタマイズ。完全なカスタマイズは、CARMA を始動するために選択した方式によって異なります。詳細については、85 ページの『CRASRV.properties、CARMA への RSE インターフェース』を参照してください。

4. 選択した CARMA 始動方式と選択した RAM に応じて、関連する構成ファイルに必要なカスタマイズを行います。詳細については、以下を参照してください。
 - 88 ページの『crastart*.conf、CRASTART サーバーの始動』
 - 92 ページの『CRASUB*、バッチ実行依頼サーバー始動』
5. オプションとして、CA Endeavor® SCM 固有の構成メンバーをカスタマイズします。詳細については、98 ページの『CRACFG、CRASHOW、および CRATMAP、CA Endeavor® SCM RAM 構成ファイル』および 100 ページの『CA Endeavor® SCM RAM バッチ・アクション』を参照してください。
6. オプションで、データ・セット割り振り exec を更新します。詳細については、99 ページの『CRANDVRA、CA Endeavor® SCM RAM 割り振り exec』、103 ページの『CRAALLOC、カスタム RAM 割り振り exec』、および 106 ページの『(オプション) カスタム割り振り exec』を参照してください。
7. オプションで、始動ユーザー出口を作成します。詳細については、(オプション) CARMA ユーザー出口を参照してください。
8. オプションとして、IRXJCL に代わる CRAXJCL を作成します。詳細については、107 ページの『(オプション) IRXJCL と CRAXJCL』を参照してください。

注: この章で参照するサンプルのメンバーは FEK.#CUST.* および /etc/rdz の中に置かれます。ただし、FEK.SFEKSAMP(FEKSETUP) ジョブをカスタマイズして実行依頼したときに、別のロケーションを指定した場合は除きます。詳しくは、20 ページの『カスタマイズのセットアップ』を参照してください。

サーバー始動方式とアクティブ RAM の選択

Developer for System z は CARMA サーバーを始動する複数の方式をサポートしています。Developer for System z も複数の Repository Access Manager (RAM) を備えています。これらは、実動 RAM とサンプル RAM という 2 つのグループに分けられます。ここでは、RAM とサーバー始動方式に考えられるいくつかの組み合わせを説明します。説明される構成シナリオは、それぞれ事前構成済みのセットアップとして使用できます。

CARMA サーバーの始動

Developer for System z は CARMA サーバーを始動する複数の方式をサポートしています。それぞれの方式には利点と欠点があります。

CRASTART

「CRASTART」方式は、CARMA サーバーを RSE 内のサブタスクとして始動します。この方式では、CARMA サーバーを始動するために必要なデータ・セット割り振りとプログラム呼び出しを別個の構成ファイルで定義し、その構成ファイルを使用するので、非常に柔軟なセットアップが可能です。この方式では最良のパフォーマンスが得られ、使用するリソースも最少で済みますが、CRASTART モジュールを LPA 内に配置する必要があります。

バッチ実行依頼

「バッチ実行依頼」方式は、ジョブを実行依頼することによって CARMA サーバーを始動します。これが、提供されたサンプル構成ファイルで使用されるデフォルトの方式です。この方式の利点は、ジョブ出力内で CARMA ログに簡単にアクセスで

きることです。また、開発者自身が保守する開発者ごとのカスタム・サーバー JCL を使用できます。ただし、この方式では、CARMA サーバーを始動する開発者ごとに JES イニシエーターが 1 つ使用されます。

(非推奨) TSO/ISPF クライアント・ゲートウェイ

「TSO/ISPF クライアント・ゲートウェイ」方式は、ISPF の TSO/ISPF クライアント・ゲートウェイを使用して TSO 環境または ISPF 環境を作成し、その中で CARMA サーバーを始動します。この方式では、ISPF.conf の機能を使用するので柔軟なデータ・セット割り振りが可能です。ただし、この方式は通常の TSO または ISPF 操作に干渉する SCM へのアクセスには適していません。

注: 「TSO/ISPF クライアント・ゲートウェイ」接続方式は、非推奨となりました。サポートはまだ行われていますが、この機能は今後拡張されず、資料は Developer for System z ライブラリーで提供されているホワイト・ペーパー「*Using ISPF Client Gateway to provide CARMA services*」(SC14-7292) に移されました (<http://www-01.ibm.com/software/awdtools/rdz/library/>)。

実動 RAM

実動タイプの RAM はすべての機能が使用できる事前組み込み型の RAM で、実稼働環境で SCM にアクセスするために使用できます。

CA Endeavor® SCM RAM

IBM Rational Developer for System z Interface for CA Endeavor® Software Configuration Manager は、Developer for System z クライアントが CA Endeavor® SCM に直接アクセスできるようにします。これ以降は、IBM Rational Developer for System z Interface for CA Endeavor® SCM を CA Endeavor® SCM RAM と略記します。

CA Endeavor® SCM パッケージ RAM

CA Endeavor® SCM パッケージ RAM は、Developer for System z クライアントが CA Endeavor® SCM パッケージに直接アクセスできるようにします。

サンプル RAM

サンプルの RAM は CARMA 環境の構成をテストする目的で、また、独自の RAM を開発するための例として提供されています。ソース・コードが組み込まれています。

重要: 提供されているサンプルの RAM を実稼働環境で使用しないでください。

PDS RAM

PDS RAM は、「リモート・システム」ビューの「MVS ファイル」->「ユーザー・データ・セット」と同様のデータ・セット・リストを提供します。

スケルトン RAM

スケルトン RAM は、ユーザーが独自の RAM を開発する際の開始点として使用できる機能フレームワークを提供します。

SCLM RAM

SCLM RAM は、ISPF の Software Configuration Manager である SCLM への基本的な入り口となります。SCLM RAM は、デフォルトでは使用可能に設定されていません。

事前構成済みの RAM とサーバー始動の組み合わせ

事前構成のセットアップとして、RAM とサーバー始動方式のいくつかの組み合わせが可能です。ご使用の環境に適合させるため、リストされたシナリオで必要なカスタマイズは、マイナーなもののみです。

- 『CA Endeavor® SCM RAM による CRAFTSTART』
- 77 ページの『サンプル RAM による CRAFTSTART』
- 79 ページの『CA Endeavor® SCM RAM によるバッチ実行依頼』
- 82 ページの『サンプル RAM によるバッチ実行依頼』

シナリオごとのさまざまなステップについて詳しくは、85 ページの『CARMA 構成詳細』を参照してください。

RAM を任意の CARMA セットアップに追加することは、現時点でも今後の時点でも可能です。既存のセットアップへの RAM の追加の詳細については、104 ページの『(オプション) 複数の RAM のサポート』を参照してください。

CA Endeavor® SCM RAM による CRAFTSTART

このシナリオの記載内容は、以下の指定で CARMA をセットアップする方法を説明したものです。

- サーバー始動: CRAFTSTART 方式。この方式では、CRAFTSTART が LPA 内にある必要があります。
- RAM: CA Endeavor® SCM RAM。

指定が異なる他のシナリオのいずれかを使用する場合、このカスタマイズ・ステップは省略できます。

CARMA VSAM データ・セットの作成

CARMA 関連の VSAM データ・セットを定義し、それらのデータ・セットにデータを設定するには、以下の JCL ジョブをカスタマイズして実行依頼します。カスタマイズの手順については、メンバー内のドキュメンテーションを参照してください。既存の VSAM データ・セットは置き換えられます。

このステップの詳細については、94 ページの『CARMA VSAM データ・セット』を参照してください。

- FEK.#CUST.JCL(CRA\$VCAD)
- FEK.#CUST.JCL(CRA\$VCAS)
- FEK.#CUST.JCL(CRA\$VMSG)

CRASRV.properties のカスタマイズ

RSE サーバーは、`/etc/rdz/CRASRV.properties` 内の設定を使用して CARMA サーバーの始動および接続を行います。このファイルは、**TSO OEDIT** コマンドで編集できます。変更を有効にするには、RSED 開始タスクを再始動してください。

デフォルトのファイル・ロケーションを使用する場合、必要な変更は、`clist.dsname` ディレクティブの値を `*CRASTART` に変更すること、および `crastart.configuration.file` の値を `/etc/rdz/crastart.endevor.conf` に変更することのみです。各種ディレクティブの詳細については、85 ページの『`CRASRV.properties`、CARMA への RSE インターフェース』を参照してください。

```
clist.dsname=*CRASTART
crastart.configuration.file=crastart.endevor.conf
```

図 15. *CRASRV.properties: CA Endeavor® SCM RAM* による *CRASTART*

crastart.endevor.conf のカスタマイズ

CRASTART は、`/etc/rdz/crastart.endevor.conf` の定義を使用して CARMA サーバーを始動するのに有効な TSO/ISPF 環境を作成します。このファイルは、**TSO OEDIT** コマンドで編集できます。変更は、更新後に始動されたすべての CARMA サーバーについて有効になります。

カスタマイズの手順については、ファイル内のドキュメンテーションを参照してください。CRASTART 始動方式の詳細については、88 ページの『`crastart*.conf`、CRASTART サーバーの始動』を参照してください。

注: ページ幅の制約上、サンプルのいくつかの行は次の行に折り返されています。インデントで始まる行はすべて、前の行の終わりに追加しなければなりません。

```

* DD used by RAM
TYPEMAP = FEK.#CUST.PARMLIB(CRATMAP)
SHOWVIEW= FEK.#CUST.PARMLIB(CRASHOW)
CRACFG = FEK.#CUST.PARMLIB(CRACFG)
* uncomment CRABCFG and CRABSKEL to use batch actions
*CRABCFG = FEK.#CUST.PARMLIB(CRABCFG)
*CRABSKEL= FEK.#CUST.CNTL
* uncomment and provide correct DSN to use Package Ship
*APIHJC = #shiphjc
CONLIB = CA.NDVR.CSIQLOAD
-COMMAND=ALLOC FI(JCLOUT) SYSOUT(A) WRITER(INTRDR) RECFM(F) LRECL(80)
  BLKSIZE(80)
-COMMAND=ALLOC FI(EXT1ELM) NEW DELETE DSORG(PS) RECFM(V,B) LRECL(4096)
  BLKSIZE(27998) SPACE(5,5) TRACKS UNIT(SYSALLDA)
-COMMAND=ALLOC FI(EXT2ELM) NEW DELETE DSORG(PS) RECFM(V,B) LRECL(4096)
  BLKSIZE(27998) SPACE(5,5) TRACKS UNIT(SYSALLDA)
-COMMAND=ALLOC FI(EXT1DEP) NEW DELETE DSORG(PS) RECFM(V,B) LRECL(4096)
  BLKSIZE(27998) SPACE(5,5) TRACKS UNIT(SYSALLDA)
C1EXMSG= SYSOUT(H)
C1MSG1 = SYSOUT(H)
MSG3FILE= DUMMY

* DD used by CARMA server (CRASERV)
* pay attention to APF authorizations when using TASKLIB
TASKLIB = FEK.SFEKLOAD,CA.NDVR.CSIQAUTH,CA.NDVR.CSIQAUTU
CRADEF = FEK.#CUST.CRADEF
CRAMSG = FEK.#CUST.CRAMSG
CRASTRS = FEK.#CUST.CRASTRS
CARMALOG= SYSOUT(H)
SYSPRINT= SYSOUT(H)

* DD used by ISPF (via NDVRC1)
-COMMAND=ALLOC FI(ISPCTL0) NEW DELETE DSORG(PS) RECFM(F,B) LRECL(80)
  BLKSIZE(32720) SPACE(5,5) TRACKS UNIT(SYSALLDA)
-COMMAND=ALLOC FI(ISPCTL1) NEW DELETE DSORG(PS) RECFM(F,B) LRECL(80)
  BLKSIZE(32720) SPACE(5,5) TRACKS UNIT(SYSALLDA)
-COMMAND=ALLOC FI(ISPPROF) NEW DELETE DSORG(PO) RECFM(F,B) LRECL(80)
  BLKSIZE(32720) SPACE(5,5) TRACKS UNIT(SYSALLDA) DIR(5)
ISPTABL = -ISPPROF
ISPTLIB = -ISPPROF,ISP.SISPTENU
ISPMLIB = ISP.SISPMENU
ISPPLIB = ISP.SISPPENU
ISPSLIB = ISP.SISPSENU

* DD used by TSO (IKJEFT01)
SYSPROC = FEK.SFEKPROC
SYSTSIN = DUMMY
SYSTSPRT= SYSOUT(H)

* CRANDVRA

PROGRAM=IKJEFT01 %CRANDVRA NDVRC1 PGM(CRASERV) PARM(&CRAPRM1.
  &CRAPRM2. &CRAPRM3. &CRAPRM4. &CRAPRM5. &CRAPRM6. &CRAPRM7.
  &CRAPRM8. )

```

図 16. crastart.endevor.conf: CA Endevor® SCM RAM による CRASTART

(オプション) 追加の CA Endevor® SCM RAM のカスタマイズ

CA Endevor® SCM RAM には、必要に応じてカスタマイズが可能な追加のコンポーネントがあります。

- CARMA 始動処理にはオプションのユーザー出口があります。詳しくは、(オプション) CARMA ユーザー出口を参照してください。

- CA Endeavor® SCM RAM には、FEK.#CUST.PARMLIB(CRACFG)、FEK.#CUST.PARMLIB(CRASHOW)、および FEK.#CUST.PARMLIB(CRATMAP) という、カスタマイズ可能な複数の構成ファイルがあります。詳しくは、98 ページの『CRACFG、CRASHOW、および CRATMAP、CA Endeavor® SCM RAM 構成ファイル』を参照してください。
- CA Endeavor® SCM RAM には割り振り exec FEK.SFEKPROC(CRANDVRA) があり、これもカスタマイズできます。詳しくは、99 ページの『CRANDVRA、CA Endeavor® SCM RAM 割り振り exec』を参照してください。
- CA Endeavor® SCM RAM は、バッチ・モードでの CA Endeavor® SCM アクションの実行をサポートしています。バッチ・アクションには構成ファイル FEK.#CUST.PARMLIB(CRABCFG) およびスケルトン JCL の FEK.#CUST.CNTL (CRABATCA) が必要で、これらはカスタマイズする必要があります。詳しくは、100 ページの『CA Endeavor® SCM RAM バッチ・アクション』を参照してください。

サンプル RAM による CRAFTSTART

このシナリオの記載内容は、以下の指定で CARMA をセットアップする方法を説明したものです。

- サーバー始動: CRAFTSTART 方式。この方式では、CRAFTSTART が LPA 内にある必要があります。
- RAM: サンプル RAM (実動環境で使用するものではありません)

異なる指定で他のシナリオのいずれかを使用したい場合は、このカスタマイズ・ステップを迂回できます。

CARMA VSAM データ・セットの作成

以下の JCL ジョブをカスタマイズして実行依頼し、CARMA 関連の VSAM データ・セットを定義して取り込みます。カスタマイズの手順については、メンバー内のドキュメンテーションを参照してください。既存の VSAM データ・セットは置き換えられます。

このステップの詳細については、94 ページの『CARMA VSAM データ・セット』および 95 ページの『CARMA Repository Access Manager (RAM)』を参照してください。

CARMA

- FEK.#CUST.JCL(CRA\$VDEF)
- FEK.#CUST.JCL(CRA\$VMSG)
- FEK.#CUST.JCL(CRA\$VSTR)

サンプル RAM

- FEK.#CUST.JCL(CRA#VPDS)

CRASRV.properties のカスタマイズ

RSE サーバーは、`/etc/rdz/CRASRV.properties` 内の設定を使用して CARMA サーバーの始動および接続を行います。このファイルは、TSO **OEDIT** コマンドで編集できます。変更を有効にするには、RSED 開始タスクを再始動する必要があります。

デフォルトのファイル・ロケーションを使用する場合、必要な変更は、`clist.dsname` ディレクティブの値を `*CRASTART` に変更することのみです。各種ディレクティブの詳細については、85 ページの『CRASRV.properties、CARMA への RSE インターフェース』を参照してください。

```
clist.dsname=*CRASTART
crastart.configuration.file=crastart.conf
```

図 17. *CRASRV.properties*: サンプル RAM による CRASTART

crastart.conf のカスタマイズ

CRASTART は、`/etc/rdz/crastart.conf` の定義を使用して、CARMA サーバーを始動するのに有効な TSO/ISPF 環境を作成します。このファイルは、TSO **OEDIT** コマンドで編集できます。変更は、更新後に始動されたすべての CARMA サーバーについて有効になります。

カスタマイズの手順については、ファイル内のドキュメンテーションを参照してください。CRASTART 始動方式の詳細については、88 ページの『`crastart*.conf`、CRASTART サーバーの始動』を参照してください。

```

* DD used by RAM
CRARAM1 = FEK.#CUST.CRARAM1                                * PDS RAM
* DD used by CARMA server (CRASERV)
TASKLIB = FEK.SFEKLOAD
CRADEF = FEK.#CUST.CRADEF
CRAMSG = FEK.#CUST.CRAMSG
CRASTRS = FEK.#CUST.CRASTRS
CARMALOG= SYSOUT(H)
SYSPRINT= SYSOUT(H)

* DD used by ISPF (ISPSTART)
-COMMAND=ALLOC FI(ISPCTL0) NEW DELETE DSORG(PS) RECFM(F,B) LRECL(80)
  BLKSIZE(32720) SPACE(5,5) TRACKS UNIT(SYSALLDA)
-COMMAND=ALLOC FI(ISPCTL1) NEW DELETE DSORG(PS) RECFM(F,B) LRECL(80)
  BLKSIZE(32720) SPACE(5,5) TRACKS UNIT(SYSALLDA)
-COMMAND=ALLOC FI(ISPPROF) NEW DELETE DSORG(PO) RECFM(F,B) LRECL(80)
  BLKSIZE(32720) SPACE(5,5) TRACKS UNIT(SYSALLDA) DIR(5)
ISPTABL = -ISPPROF
ISPTLIB = -ISPPROF,ISP.SISPTENU
ISPLIB = ISP.SISPMENU
ISPPLIB = ISP.SISPPENU
ISPSLIB = ISP.SISPSENU

* DD used by TSO (IKJEFT01)
SYSPROC = #hlq.SFEKPROC                                      * CRAALLOC
SYSTSIN = DUMMY
SYSTSPRT= SYSOUT(H)

PROGRAM=IKJEFT01 %CRAALLOC ISPSTART PGM(CRASERV) PARM(&CRAPRM1.
  &CRAPRM2. &CRAPRM3. &CRAPRM4. &CRAPRM5. &CRAPRM6. &CRAPRM7.
  &CRAPRM8. )

```

図 18. *crastart.conf*: サンプル RAM による *CRASTART*

注: ページ幅の制約上、サンプルのいくつかの行は次の行に折り返されています。インデントで始まる行はすべて、前の行の終わりに追加しなければなりません。

(オプション) 追加のカスタム RAM のカスタマイズ

カスタム RAM には、必要に応じてカスタマイズが可能な追加のコンポーネントがあります。

- CARMA 始動処理にはオプションのユーザー出口があります。詳しくは、(オプション) CARMA ユーザー出口を参照してください。
- カスタム RAM 始動には、割り振り `exec FEK.SFEKPROC(CRAALLOC)` があり、これもカスタマイズできます。詳しくは、CRAALLOC、カスタム RAM 割り振り `exec` を参照してください。

CA Endeavor® SCM RAM によるバッチ実行依頼

このシナリオの記載内容は、以下の指定で CARMA をセットアップする方法を説明したものです。

- サーバー始動: バッチ実行依頼方式。この方式には JES イニシエーターが必要です。
- RAM: CA Endeavor® SCM RAM。

指定が異なる他のシナリオのいずれかを使用する場合、このカスタマイズ・ステップは省略できます。

CARMA VSAM データ・セットの作成

以下の JCL をカスタマイズして実行依頼し、CARMA 関連の VSAM データ・セットを定義して取り込みます。カスタマイズの手順については、メンバー内のドキュメンテーションを参照してください。既存の VSAM データ・セットは置き換えられます。

このステップの詳細については、94 ページの『CARMA VSAM データ・セット』を参照してください。

- FEK.#CUST.JCL(CRA\$VCAD)
- FEK.#CUST.JCL(CRA\$VCAS)
- FEK.#CUST.JCL(CRA\$VMSG)

CRASRV.properties のカスタマイズ

RSE サーバーは、/etc/rdz/CRASRV.properties 内の設定を使用して CARMA サーバーの始動および接続を行います。このファイルは、TSO **OEDIT** コマンドで編集できます。変更を有効にするには、RSED 開始タスクを再始動する必要があります。

デフォルトのファイル・ロケーションを使用する場合、必要な変更は、clist.dsname ディレクティブの値を FEK.#CUST.CNTL(CRASUBCA) に変更することのみです。各種ディレクティブの詳細については、85 ページの『CRASRV.properties、CARMA への RSE インターフェース』を参照してください。

```
clist.dsname='FEK.#CUST.CNTL(CRASUBCA)'
```

図 19. CRASRV.properties - CA Endeavor® SCM RAM によるバッチ実行依頼

CRASUBCA のカスタマイズ

FEK.#CUST.CNTL(CRASUBCA) CLIST と組み込み JCL が CARMA サーバーに実行依頼します。変更は、更新後に始動されたすべての CARMA サーバーについて有効になります。

カスタマイズの手順については、メンバー内のドキュメンテーションを参照してください。バッチ実行依頼始動方式の詳細については、92 ページの『CRASUB*、バッチ実行依頼サーバー始動』を参照してください。

```

PROC 8 CRAPRM1 CRAPRM2 CRAPRM3 CRAPRM4 CRAPRM5 CRAPRM6 CRAPRM7 CRAPRM8
SUBMIT * END($$)
//CRA&PORT JOB CLASS=A,MSGCLASS=A,MSGLEVEL=(1,1)
//*
//RUN      EXEC PGM=IKJEFT01,DYNAMNBR=125,REGION=0M,TIME=NOLIMIT
//*
//* DD used by RAM
//TYPEMAP DD DISP=SHR,DSN=FEK.#CUST.PARMLIB(CRATMAP)
//SHOWVIEW DD DISP=SHR,DSN=FEK.#CUST.PARMLIB(CRASHOW)
//CRACFG DD DISP=SHR,DSN=FEK.#CUST.PARMLIB(CRACFG)
//* uncomment CRABCFG and CRABSKEL to use batch actions
//*CRABCFG DD DISP=SHR,DSN=FEK.#CUST.PARMLIB(CRABCFG)
//*CRABSKEL DD DISP=SHR,DSN=FEK.#CUST.CNTL
//* uncomment and provide correct DSN to use Package Ship
//*APIHJC DD DISP=SHR,DSN=#shipjc
//CONLIB DD DISP=SHR,DSN=CA.NDVR.CSIQLOAD
//JCLOUT DD SYSOUT=(A,INTRDR),DCB=(LRECL=80,RECFM=F,BLKSIZE=80)
//EXT1ELM DD DISP=(NEW,DELETE),UNIT=SYSALLDA,
//          RECFM=VB,LRECL=4096,BLKSIZE=27998,SPACE=(TRK,(5,5))
//EXT2ELM DD DISP=(NEW,DELETE),UNIT=SYSALLDA,
//          RECFM=VB,LRECL=4096,BLKSIZE=27998,SPACE=(TRK,(5,5))
//EXT1DEP DD DISP=(NEW,DELETE),UNIT=SYSALLDA,
//          RECFM=VB,LRECL=4096,BLKSIZE=27998,SPACE=(TRK,(5,5))
//C1MSG1 DD SYSOUT(H)
//C1EXMSG1 DD SYSOUT(H)
//MSG3FILE DD DUMMY
//*
//* DD used by CARMA server (CRASERV)
//* pay attention to APF authorizations when using STEPLIB
//STEPLIB DD DISP=SHR,DSN=FEK.SFEKLOAD
//          DD DISP=SHR,DSN=CA.NDVR.CSIQAUTH
//          DD DISP=SHR,DSN=CA.NDVR.CSIQAUTU
//CRADEF DD DISP=SHR,DSN=FEK.#CUST.CRADEF
//CRAMSG DD DISP=SHR,DSN=FEK.#CUST.CRAMSG
//CRASTRS DD DISP=SHR,DSN=FEK.#CUST.CRASTRS
//CARMALOG DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//*
//* DD used by ISPF (via NDVRC1)
//ISPPROF DD DISP=(NEW,DELETE,DELETE),UNIT=SYSALLDA,
//          LRECL=80,RECFM=FB,SPACE=(TRK,(1,1,5))
//ISPCTL0 DD DISP=(NEW,DELETE,DELETE),UNIT=SYSALLDA,
//          LRECL=80,RECFM=FB,SPACE=(TRK,(5,5))
//ISPCTL1 DD DISP=(NEW,DELETE,DELETE),UNIT=SYSALLDA,
//          LRECL=80,RECFM=FB,SPACE=(TRK,(5,5))
//ISPLIB DD DISP=SHR,DSN=ISP.SISPMENU
//ISPPLIB DD DISP=SHR,DSN=ISP.SISPPENU
//ISPSLIB DD DISP=SHR,DSN=ISP.SISPSENU
//ISPTLIB DD DISP=SHR,DSN=ISP.SISPTENU
//*
//* DD used by TSO (IKJEFT01)
//SYSPROC DD DISP=SHR,DSN=FEK.SFEKPROC
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
%CRANDVRA NDVRC1 PGM(CRASERV) PARM(&CRAPRM1 &CRAPRM2 &STR(+))
&CRAPRM3 &STR(+)
&CRAPRM4 &STR(+)
&CRAPRM5 &STR(+)
&CRAPRM6 &STR(+)
&CRAPRM7 &STR(+)
&CRAPRM8 &STR(+) )
$$
EXIT CODE(0)

```

図 20. CRASUBCA: CA Endeavor® SCM RAM によるバッチ実行依頼

(オプション) 追加の CA Endeavor® SCM RAM のカスタマイズ

CA Endeavor® SCM RAM には、必要に応じてカスタマイズが可能な追加のコンポーネントがあります。

- CA Endeavor® SCM RAM には、FEK.#CUST.PARMLIB(CRACFG)、FEK.#CUST.PARMLIB(CRASHOW) および FEK.#CUST.PARMLIB(CRATMAP) という、カスタマイズ可能な複数の構成ファイルがあります。詳しくは、98 ページの『CRACFG、CRASHOW、および CRATMAP、CA Endeavor® SCM RAM 構成ファイル』を参照してください。
- CA Endeavor® SCM RAM には割り振り exec FEK.SFEKPROC(CRANDVRA) があり、これもカスタマイズできます。詳しくは、99 ページの『CRANDVRA、CA Endeavor® SCM RAM 割り振り exec』を参照してください。
- CA Endeavor® SCM RAM は、バッチ・モードでの CA Endeavor® SCM アクションの実行をサポートしています。バッチ・アクションには構成ファイル FEK.#CUST.PARMLIB(CRABCFG)、スケルトン JCL の FEK.#CUST.CNTL(CRABATCA)、およびオプションのデフォルト・ジョブ・カード FEK.#CUST.CNTL(CRABJOBC) が必要で、これらはカスタマイズする必要があります。詳しくは、100 ページの『CA Endeavor® SCM RAM バッチ・アクション』を参照してください。

サンプル RAM によるバッチ実行依頼

このシナリオの記載内容は、以下の指定で CARMA をセットアップする方法を説明したものです。

- サーバー始動: バッチ実行依頼方式 (JES イニシエーターが必要)
- RAM: サンプル RAM (実動環境で使用するものではありません)

指定が異なる他のシナリオのいずれかを使用する場合、このカスタマイズ・ステップは省略できます。

VSAM データ・セットの作成

以下の JCL ジョブをカスタマイズして実行依頼し、CARMA 関連の VSAM データ・セットを定義して取り込みます。カスタマイズの手順については、メンバー内のドキュメンテーションを参照してください。既存の VSAM データ・セットは置き換えられます。

このステップの詳細については、94 ページの『CARMA VSAM データ・セット』および 95 ページの『CARMA Repository Access Manager (RAM)』を参照してください。

CARMA

- FEK.#CUST.JCL(CRA\$VDEF)
- FEK.#CUST.JCL(CRA\$VMSG)
- FEK.#CUST.JCL(CRA\$VSTR)

サンプル RAM

- FEK.#CUST.JCL(CRA#VPDS)

CRASRV.properties のカスタマイズ

RSE サーバーは、`/etc/rdz/CRASRV.properties` 内の設定を使用して CARMA サーバーの始動および接続を行います。このファイルは、**TSO OEDIT** コマンドで編集できます。変更を有効にするには、RSED 開始タスクを再始動する必要があります。

これは Developer for System z のデフォルトのシナリオであるため、変更を加えていないファイルのコピーから開始する場合、変更の必要はありません。各種ディレクトタイプの詳細については、85 ページの『CRASRV.properties、CARMA への RSE インターフェース』を参照してください。

```
clist.dsname='FEK.#CUST.CNTL(CRASUBMT)'
```

図 21. *CRASRV.properties*: サンプル RAM によるバッチ実行依頼

CRASUBMT のカスタマイズ

`FEK.#CUST.CNTL(CRASUBMT)` CLIST と組み込み JCL が CARMA サーバーに実行依頼します。変更は、更新後に始動されたすべての CARMA サーバーについて有効になります。

カスタマイズの手順については、メンバー内のドキュメンテーションを参照してください。バッチ実行依頼始動方式の詳細については、92 ページの『CRASUB*、バッチ実行依頼サーバー始動』を参照してください。

```

PROC 8 CRAPRM1 CRAPRM2 CRAPRM3 CRAPRM4 CRAPRM5 CRAPRM6 CRAPRM7 CRAPRM8
SUBMIT * END($$)
//CRA&PORT JOB CLASS=A,MSGCLASS=A,MSGLEVEL=(1,1)
//*
//RUN      EXEC PGM=IKJEFT01,DYNAMNBR=125,REGION=0M,TIME=NOLIMIT
//*
//* DD used by RAM
//CRARAM1 DD DISP=SHR,DSN=FEK.#CUST.CRARAM1          * PDS RAM
//*
//* DD used by CARMA server (CRASERV)
//STEPLIB DD DISP=SHR,DSN=FEK.SFEKLOAD
//CRADEF  DD DISP=SHR,DSN=FEK.#CUST.CRADEF
//CRAMSG  DD DISP=SHR,DSN=FEK.#CUST.CRAMSG
//CRASTRS DD DISP=SHR,DSN=FEK.#CUST.CRASTRS
//CARMALOG DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//*
//* DD used by ISPF (ISPSTART)
//ISPPROF DD DISP=(NEW,DELETE,DELETE),UNIT=SYSALLDA,
//          LRECL=80,RECFM=FB,SPACE=(TRK,(1,1,5))
//ISPCTL0 DD DISP=(NEW,DELETE,DELETE),UNIT=SYSALLDA,
//          LRECL=80,RECFM=FB,SPACE=(TRK,(5,5))
//ISPCTL1 DD DISP=(NEW,DELETE,DELETE),UNIT=SYSALLDA,
//          LRECL=80,RECFM=FB,SPACE=(TRK,(5,5))
//ISPLIB  DD DISP=SHR,DSN=ISP.SISPMENU
//ISPPLIB DD DISP=SHR,DSN=ISP.SISPPENU
//ISPSLIB DD DISP=SHR,DSN=ISP.SISPSENU
//ISPTLIB DD DISP=SHR,DSN=ISP.SISPTENU
//*
//* DD used by TSO (IKJEFT01)
//SYSPROC DD DISP=SHR,DSN=#hlq.SFEKPROC              * CRAALLOC
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
%CRAALLOC ISPSTART PGM(CRASERV) PARM(&CRAPRM1 &CRAPRM2 &STR(+))
&CRAPRM3 &STR(+)
&CRAPRM4 &STR(+)
&CRAPRM5 &STR(+)
&CRAPRM6 &STR(+)
&CRAPRM7 &STR(+)
&CRAPRM8 &STR(+)
)
$$
EXIT CODE(0)

```

図 22. CRASUBMT: サンプル RAM によるバッチ実行依頼

(オプション) 追加のカスタム RAM のカスタマイズ

カスタム RAM には、必要に応じてカスタマイズが可能な追加のコンポーネントがあります。

- CARMA 始動処理にはオプションのユーザー出口があります。詳しくは、(オプション) CARMA ユーザー出口を参照してください。
- カスタム RAM 始動には、割り振り `exec FEK.SFEKPROC(CRAALLOC)` があり、これもカスタマイズできます。詳しくは、CRAALLOC、カスタム RAM 割り振り `exec` を参照してください。

CARMA 構成詳細

本書に記載のさまざまな構成シナリオでは、多数の CARMA 構成ファイルを共有しています。これら構成ファイルの詳細をここに記載しますが、これらは、さまざまなシナリオ内部から参照されています。

CRASRV.properties、CARMA への RSE インターフェース

CARMA サーバーは、ホスト・システムを使用して 1 つ以上の Software Configuration Manager (SCM) にアクセスする他の製品のために標準 API を提供します。しかし、クライアント・コンピューターと直接通信するための方式は提供しません。このような通信については、RSE サーバーなどの他の製品に依存しています。RSE サーバーは、CRASRV.properties 内の設定を使用して CARMA サーバーの始動および接続を行います。

CRASRV.properties は /etc/rdz/ に置かれます。ただし、FEK.SFEKSAMP(FEKSETUP) ジョブをカスタマイズして実行依頼したときに、別のロケーションを指定した場合は除きます。詳しくは、20 ページの『カスタマイズのセットアップ』を参照してください。このファイルは、TSO **OEDIT** コマンドで編集できます。

注: 変更を有効にするには、RSED 開始タスクを再始動する必要があります。

```
# CRASRV.properties - CARMA configuration options
#
port.start=0
#port.range=100
#user.exit='FEK.SFEKSAMP(CRAEXIT)'
startup.script.name=carma.startup.rex
clist.dsname='FEK.#CUST.CNTL(CRASUBMT)'
crastart.configuration.file=crastart.conf
#crastart.stub=/usr/lpp/rdz/bin/CRASTART
#crastart.syslog=Partial
#crastart.timeout=420
#crastart.steplib=FEK.SFEKLPA
#crastart.tasklib=TASKLIB
```

図 23. CRASRV.properties - CARMA 構成ファイル

port.start

port.start の値が 0 (ゼロ) である場合、CARMA は、CARMA と RSE サーバーの間の通信に一時ポートを使用します。このシナリオでは、TCP/IP はランダムな空きポート番号を割り当てます。port.start の値がゼロ以外の場合、その値は、CARMA と RSE サーバーの間の通信に使用されるポート範囲の開始点として解釈されます。この場合、port.range 変数も定義する必要があります。デフォルトのポートは 0 です。このポート上の通信は、ご使用のホスト・システムだけに限定されます。

注: ポートを選択する前に、**NETSTAT** コマンドおよび **NETSTAT PORTL** コマンドを使用して、そのポートがシステム上で使用可能であることを確認してください。詳しくは、「ホスト構成リファレンス」(SA88-4226) の『予約済み TCP/IP ポート』を参照してください。

#port.range

port.start がゼロ以外の場合に CARMA 通信に使用される、port.start から始まるポートの範囲。デフォルトは 100 です。ポート範囲のサイズを

指定するには、コメントを外してカスタマイズします。例えば、`port.start` が 5227 で、`port.range` が 100 の場合、CARMA はポート 5227 から 5326 (両端の値を含む) までを使用できます。各 CARMA 接続でポートが排他的に使用されるため、ポート範囲を指定することで同時 CARMA セッションの最大数が制限されます。

#user.exit

CARMA 始動時に実行されるユーザー指定コードを定義します。コメントを外し、実行するコードのデータ・セット名を指定します。

引用符 (') を付けた場合、そのデータ・セット名は絶対参照となり、引用符 (') を付けなかった場合、そのデータ・セット名には TSO 接頭部ではなく、クライアント・ユーザー ID が接頭部として付加されます。後者の場合は、すべての CARMA ユーザーが独自の出口コードを保守する必要があります。

SFEKSAMP(CRAEXIT) というサンプルのユーザー出口が提供されています。このサンプルでは、ユーザー出口に渡される始動引数について説明しています。詳しくは、(オプション) CARMA ユーザー出口を参照してください。

startup.script.name

CARMA 始動スクリプトを定義します。デフォルトは `carma.startup.rex` です。この REXX exec は CARMA サーバーの始動をトリガーします。ファイル名はいくつかの方法で指定できます。

- ヌル・ストリング。これは変数が指定されていないことを意味します。この場合、デフォルト値が使用されます。
- ファイル名のみ。これがデフォルトの方式です。CARMA は、PATH 環境変数内のディレクトリーを検索してファイルを見付けます。Developer for System z 実行可能ファイルを保持しているディレクトリー (デフォルトでは `/usr/lpp/rdz/bin`) は自動的に PATH 環境変数に追加されます。
- 相対パス (ディレクトリーやファイル名などで先頭がスラッシュ (/) から始まらないもの)。CARMA が構成ディレクトリー (デフォルトでは `/etc/rdz/`) を提供されたパスに追加して絶対パスにします。
- 絶対パス (ディレクトリーやファイル名などで先頭がスラッシュ (/) から始まるもの)。CARMA は、指定されたファイル・ロケーションを使用します。

clist.dsname

CARMA サーバーの始動方式を定義します。各種始動方式の詳細については、72 ページの『サーバー始動方式とアクティブ RAM の選択』を参照してください。

- *CRASTART は、CRASTART を使用して、CARMA サーバーを RSE 内のサブタスクとして始動するよう指示します。*CRASTART を指定する場合は、`crastart.*` ディレクティブも指定する必要があります。指定しない場合、デフォルト値が使用されます。
- *ISPF は、ISPF の TSO/ISPF クライアント・ゲートウェイを使用して CARMA サーバーを始動するよう指示します。この始動方式は推奨されなくなりました。
- それ以外のすべての値は、TSO のような命名規則を使用して、CRASUBMT CLIST のロケーションを定義します。単一引用符 (') を付けた場合、そ

のデータ・セット名は絶対参照となり、単一引用符 (') を付けなかった場合、そのデータ・セット名には TSO 接頭部ではなく、クライアントのユーザー ID が接頭部として付加されます。後者の場合は、すべての CARMA ユーザーが独自の CRASUBMT CLIST を保守する必要があります。

デフォルトは 'FEK.#CUST.CNTL(CRASUBMT)' です。この CLIST は、バッチ実行依頼方式で接続を開いたときに CARMA サーバーを始動します。

crastart.configuration.file

CRASTART 構成ファイルの名前を指定します。デフォルトは `crastart.conf` です。このファイルは、CARMA サーバーを始動するために必要なデータ・セット割り振りとプログラム呼び出しを指定します。このディレクティブは、`clist.dsname` ディレクティブの値が *CRASTART である場合にのみ使用されます。ファイル名はいくつかの方法で指定できます。

- ヌル・ストリング。これは変数が指定されていないことを意味します。デフォルト値が使用されます。
- ファイル名のみ。これがデフォルトの方式です。CARMA は、構成ディレクトリー (デフォルトでは `/etc/rdz`) を検索してファイルを見付けます。
- 相対パス (ディレクトリーやファイル名などで先頭がスラッシュ (/) から始まらないもの)。CARMA が構成ディレクトリー (デフォルトでは `/etc/rdz/`) を提供されたパスに追加して絶対パスにします。
- 絶対パス (ディレクトリーやファイル名などで先頭がスラッシュ (/) から始まるもの)。CARMA は、指定されたファイル・ロケーションを使用します。

#crastart.stub

CRASTART を呼び出すための z/OS UNIX スタブ。デフォルトは CRASTART です。このスタブは、MVS ベースの CRASTART ロード・モジュールを z/OS UNIX プロセスから使用できるようにします。特定のパスを指定するには、コメントを外してカスタマイズします。このディレクティブは、`clist.dsname` ディレクティブの値が *CRASTART である場合にのみ使用されます。ファイル名はいくつかの方法で指定できます。

- ヌル・ストリング。これは変数が指定されていないことを意味します。デフォルト値が使用されます。
- ファイル名のみ。これがデフォルトの方式です。CARMA は、PATH 環境変数内のディレクトリーを検索してファイルを見付けます。Developer for System z 実行可能ファイルを保持しているディレクトリー (デフォルトでは `/usr/lpp/rdz/bin`) は自動的に PATH 環境変数に追加されます。
- 相対パス (ディレクトリーやファイル名などで先頭がスラッシュ (/) から始まらないもの)。CARMA が構成ディレクトリー (デフォルトでは `/etc/rdz/`) を提供されたパスに追加して絶対パスにします。
- 絶対パス (ディレクトリーやファイル名などで先頭がスラッシュ (/) から始まるもの)。CARMA は、指定されたファイル・ロケーションを使用します。

#crastart.syslog

CRASTART で CARMA サーバーを始動するときに、どれくらいの量の情報をシステム・ログに書き込むかを指定します。デフォルトは Partial です。有効な値を以下の表に示します。

A (All)	すべてのトレース情報を SYSLOG に出力します。
P (Partial)	接続、切断、およびエラー情報のみを SYSLOG に出力します。
それ以外のすべて	エラー条件のみを SYSLOG に出力します。

システム・ログ・メッセージに必要な詳細レベルを指定するには、コメントを外してカスタマイズします。このディレクティブは、clist.dsname ディレクティブの値が *CRASTART である場合にのみ使用されます。

#crastart.timeout

アクティビティーがないために CARMA サーバーが終了するまでの時間の長さ (秒単位)。デフォルトは 420 (7 分) です。必要なタイムアウト値を指定するには、コメントを外してカスタマイズします。このディレクティブは、clist.dsname ディレクティブの値が *CRASTART である場合にのみ使用されます。

注: SMFPRMxx parmlib メンバー内の JWT パラメーターの設定が CRASRV.properties の crastart.timeout 値より小さい場合は、モジュール CRASERV のシステム異常終了 522 が発生します。このオカレンスによって、CARMA の操作が影響を受けることはありません。必要であれば、サーバーが自動的に再始動されるためです。

#crastart.steplib

rsed.envvars 内の STEPLIB ディレクティブを通じてアクセスする場合の CRASTART モジュールのロケーション。デフォルトは FEK.SFEKLPA です。CRASTART モジュールを LPA および LINKLIST のいずれにも含めることができない場合は、このディレクティブのコメントを外してカスタマイズします。CRASTART モジュールが LPA 内にない場合は、プログラム制御および APF の問題が起きる可能性があります。このディレクティブは、clist.dsname ディレクティブの値が *CRASTART である場合にのみ使用されます。

#crastart.tasklib

crastart.conf 内の TASKLIB DD 名の代替名。デフォルトは TASKLIB です。使用している SCM または RAM にとって DD 名 TASKLIB が特殊な意味を持ち、STEPLIB の代わりとして使用できない場合は、このディレクティブのコメントを外してカスタマイズします。このディレクティブは、clist.dsname ディレクティブの値が *CRASTART である場合にのみ使用されます。

crastart*.conf、CRASTART サーバーの始動

RSE は CRASTART ロード・モジュールを始動します。このモジュールは crastart*.conf の定義を使用して、TSO パッチ・コマンドおよび ISPF パッチ・コマンドを実行するための有効な環境を作成します。Developer for System z は、その環境を使用して CARMA サーバー CRASERV を実行します。

crastart*.conf は /etc/rdz/ に置かれます。ただし、ジョブ FEK.SFEKSAMP(FEKSETUP) をカスタマイズして実行依頼したときに、別のロケーションを指定した場合は除きます。詳しくは、20 ページの『カスタマイズのセットアップ』を参照してください。このファイルは、TSO **OEDIT** コマンドで編集できます。

注: 変更は、更新後に始動されたすべての CARMA サーバーについて有効になります。

Developer for System z は、複数の crastart*.conf 構成ファイルを提供しています。これらサンプル・ファイルはそれぞれ、特定のカスタマイズ・シナリオ用に事前構成されています。

- crastart.endevor.conf は、CA Endeavor® SCM RAM による CRASTART の始動用に構成されています。
- crastart.conf は、サンプル RAM による CRASTART の始動用に構成されています。

crastart*.conf ファイルの機能は、概念上は JCL ジョブ・ストリームに似ていますが、動作はもっと限定的です。

- 以下のサンプルに、有効な行形式を示します。

```
- * comment
- ddname=dsn1,dsn2,dsn3          * comment
- ddname=SYSOUT(c)                * comment
- ddname=DUMMY                    * comment
- -COMMAND=<any bpxwdyn command> * comment
- PROGRAM = progname parms       * comment
```

注: **BPXWDYN** コマンドは、「*REXX* および *z/OS UNIX* システム・サービスの使い方」(SA88-8644) に記載されており、複雑な割り振り構成が可能です。

- すべての入力は大文字に変更されます。
- 行の継続はサポートされていません。
- 行の長さに制限はありません。
- 等号 (=) の前後に 1 つ以上のブランク・スペースが許可されます。
- DD 割り振りは、関連する PROGRAM ステートメントの前になければなりません。
- ここで割り振られる DD 名は、プログラム実行の終わりで解放されます。これらは累積されません。
- 呼び出されたプログラムにより割り振られる DD 名は解放されません。
- 複数のデータ・セットを DD 名に連結できます。データ・セット名はコンマ (,) で区切る必要があり、連結はリストされた順序で検索されます。
- すべてのデータ・セット割り振りは **DISP=SHR** で行われますが、**-COMMAND** を使用して行われる割り振りを除きます。
- インライン・データはサポートされません。すべてのデータがカタログされたファイルに入っている必要があります。
- 変数は、等号 (=) の右側でしか使用できません。
- 以下の変数がサポートされています。

&CRAUSER.	クライアント・ユーザー ID
&CRADATE.	Dyyyyddd 形式 (7 文字のユリウス日付) の現在日付。
&CRATIME.	Thhmmss 形式 (時分秒) の現在時刻。
&CRAPRM1.	ポート番号
&CRAPRM2.	タイムアウト値
システム・シンボル	任意の SYS1.PARMLIB(IEASYMxx) システム・シンボル
-<ddname>	前に定義した DD 名の前にハイフン (-) を付加すると、JCL の *.ddname 逆方向参照のように機能します。オリジナルの DD は、-COMMAND ステートメントを使用して割り振る必要があります。

注: TSO 接頭部の変数は存在しません。TSO は構成ファイルが解釈される時、アクティブではないからです。TSO 接頭部または使用できないその他の変数が必要な場合は、106 ページの『(オプション) カスタム割り振り exec』を参照してください。

図 24 は、ISPF サービスを含む基本 crastart*.conf スケルトンを示しています。

```
* DD used by RAM

* DD used by CARMA server (CRASERV)
TASKLIB = FEK.SFEKLOAD
CRADEF  = FEK.#CUST.CRADEF
CRAMSG  = FEK.#CUST.CRAMSG
CRASTRS = FEK.#CUST.CRASTRS
CARMALOG= SYSOUT(H)
SYSPRINT= SYSOUT(H)

* DD used by ISPF (ISPSTART)
-COMMAND=ALLOC FI(ISPCTL0) NEW DELETE DSORG(PS) RECFM(F,B) LRECL(80)
  BLKSIZE(32720) SPACE(5,5) TRACKS UNIT(SYSALLDA)
-COMMAND=ALLOC FI(ISPCTL1) NEW DELETE DSORG(PS) RECFM(F,B) LRECL(80)
  BLKSIZE(32720) SPACE(5,5) TRACKS UNIT(SYSALLDA)
-COMMAND=ALLOC FI(ISPPROF) NEW DELETE DSORG(PO) RECFM(F,B) LRECL(80)
  BLKSIZE(32720) SPACE(5,5) TRACKS UNIT(SYSALLDA) DIR(5)
ISPTABL = -ISPPROF
ISPTLIB = -ISPPROF,ISP.SISPTENU
ISPMLIB = ISP.SISPMENU
ISPPLIB = ISP.SISPPENU
ISPSLIB = ISP.SISPSENU

* DD used by TSO (IKJEFT01)
SYSPROC = #hlq.SFEKPROC
SYSTSIN = DUMMY
SYSTSPRT= SYSOUT(H)

* CRAALLOC

PROGRAM=IKJEFT01 %CRAALLOC ISPSTART PGM(CRASERV) PARM(&CRAPRM1.
  &CRAPRM2. &CRAPRM3. &CRAPRM4. &CRAPRM5. &CRAPRM6. &CRAPRM7.
  &CRAPRM8. )
```

図 24. crastart*.conf: CRASTART を使用した CARMA サーバーの始動

注:

- ページ幅の制約上、サンプルのいくつかの行は次の行に折り返されています。インデントで始まる行はすべて、前の行の終わりに追加しなければなりません。

- PROGRAM 行を変更する場合は、ストリングの処理を簡素化するために、`PARM()` ステートメントの右丸括弧 (“)”) の前に少なくとも 1 つの空白を必ず置いてください。
- CARMA TSO 環境をカスタマイズするために、独自の DD ステートメントとデータ・セット連結を追加し、TSO ログオン・プロシーチャーを模倣することができます。
- DD 名 TASKLIB は、JCL の STEPLIB の役割を果たします。この DD 名は、CRASRV.properties の `crastart.tasklib` に指定された値に一致している必要があります。これについては、85 ページの『CRASRV.properties、CARMA への RSE インターフェース』で説明します。
- TASKLIB 割り振りには通常の APF 規則が当てはまります。連結の一部に非 APF 許可ライブラリーがあると、ライブラリーは APF 許可を失います。
- SMFPRMxx parmlib メンバー内の JWT パラメーターの設定が CRASRV.properties の `crastart.timeout` 値より小さい場合は、モジュール CRASERV のシステム異常終了 522 が発生します。システム異常終了によって CARMA の操作が影響を受けることはありません。必要であればサーバーが自動的に再始動するためです。
- サーバーの終了時に、CARMA サーバー始動の詳細は、`rsecomm.log` に示されます。`rsecomm.log` の詳細レベルの設定について詳しくは、135 ページの『(オプション) `rsecomm.properties`、RSE トレース』を参照してください。

CRASTART ログ・ファイルの収集

CRASTART は、TSO 環境を RSE の子プロセスとして作成します。これは、別個のアドレス・スペースとして稼働します。CARMA 出力を `SYSOUT(*)` に送信し続けるために、簡単ではないアクションが必要になる場合があります。このことにより、ログ・ファイルの収集が複雑になります。この問題は、以下の割り振りの例に示すように、ログ・ファイルをユーザー固有のデータ・セットに書き込むことで解決します。

```
-COMMAND=ALLOC FI(CARMALOG) MOD CATALOG DSORG(PS) RECFM(F,B) LRECL(133)
          BLKSIZE(27930) SPACE(5,5) TRACKS UNIT(SYSALLDA)
          DA(&CRAUSER..&SYSNAME..CRA.CARMALOG)
```

注:

- ページ幅の制約上、サンプルのいくつかの行は次の行に折り返されています。インデントで始まる行はすべて、前の行の終わりに追加しなければなりません。
- ユーザー固有のログ・ファイルを作成できるようにするには、このログ・ファイルを `-COMMAND` ステートメントを使用して割り振る必要があります。
- 柔軟性を高める必要がある場合、例えばログを特定のユーザーのデータ・セットにのみ送信する場合など、ログ・データ・セットを割り振り `exec` に割り振ることもできます。割り振り `exec` の詳細については、106 ページの『(オプション) カスタム割り振り `exec`』を参照してください。

ログ・ファイルを `SYSOUT` に書き込む場合、z/OS UNIX プロセスによって割り振られた `SYSOUT` は、JES では特殊な出力として扱われることに留意してください。これは、APPC トランザクションによって割り振られた `SYSOUT` に似ています。

- CARMA サーバーがまだアクティブなときに、SDSF の **DA** コマンドを使用して出力を確認できます。ジョブには、ユーザーのユーザー ID の後にジョブ名としてのランダムな 1 桁の数字と、STC ジョブ ID を続けたものが与えられます。ユーザーは、ジョブ所有者になります。
- 非アクティブ状態またはユーザーによる接続の終了が原因で CARMA サーバーが終了する際に、出力が HOLD 出力クラスに書き込まれた場合は、SDSF の **APPC ON** コマンドおよび **H ALL** コマンドを使用して出力を確認できます。ジョブ名、ジョブ ID、およびジョブ所有者は同じままです。各 DD は個別のスプール・ファイルとして表示され、どの DD であるかは示されません。
- FEJJCNF で SEARCHALL=ON がアクティブであり、出力が HOLD 出力クラスのスプール上にある場合にも、JES ジョブ・モニターは出力を表示できます。SEARCHALL ディレクティブの詳細については、38 ページの『FEJJCNF、JES ジョブ・モニター構成ファイル』を参照してください。

CRASUB*、バッチ実行依頼サーバー始動

RSE は CLIST CRASUB* を始動します。すると、始動された CRASUB* が、組み込み JCL を実行依頼して、TSO バッチ・コマンドおよび ISPF バッチ・コマンドを実行するための有効な環境を作成します。Developer for System z は、その環境を使用して CARMA サーバー CRASERV を実行します。

CRASUB* は FEK.#CUST.CNTL に置かれます。ただし、FEK.SFEKSAMP(FEKSETUP) ジョブをカスタマイズして実行依頼したときに、別のロケーションを指定した場合は除きます。詳しくは、20 ページの『カスタマイズのセットアップ』を参照してください。

注: 変更は、更新後に始動されたすべての CARMA サーバーについて有効になります。

Developer for System z は、複数の CRASUB* JCL ジョブを提供します。これらサンプル・ファイルはそれぞれ、特定のカスタマイズ・シナリオ用に事前構成されています。

- CRASUBCA は、CA Endeavor[®] SCM RAM によるバッチ始動用に構成されています。
- CRASUBMT は、サンプル RAM によるバッチ始動用に構成されています。

93 ページの図 25 は、ISPF サービスを含む基本 CRASUB* スケルトンを示しています。

```

PROC 8 CRAPRM1 CRAPRM2 CRAPRM3 CRAPRM4 CRAPRM5 CRAPRM6 CRAPRM7 CRAPRM8
/* SET CRAPRM2=420
SUBMIT * END($$)
//CRA&PORT JOB CLASS=A,MSGCLASS=A,MSGLEVEL=(1,1)
//*
//RUN      EXEC PGM=IKJEFT01,DYNAMNBR=125,REGION=0M,TIME=NOLIMIT
//*
//* DD used by RAM
//*
//* DD used by CARMA server (CRASERV)
//STEPLIB DD DISP=SHR,DSN=FEK.SFEKLOAD
//CRADEF  DD DISP=SHR,DSN=FEK.#CUST.CRADEF
//CRAMSG  DD DISP=SHR,DSN=FEK.#CUST.CRAMSG
//CRASTRS DD DISP=SHR,DSN=FEK.#CUST.CRASTRS
//CARMALOG DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//*
//* DD used by ISPF (ISPSTART)
//ISPPROF DD DISP=(NEW,DELETE,DELETE),UNIT=SYSALLDA,
//          LRECL=80,RECFM=FB,SPACE=(TRK,(1,1,5))
//ISPCTL0 DD DISP=(NEW,DELETE,DELETE),UNIT=SYSALLDA,
//          LRECL=80,RECFM=FB,SPACE=(TRK,(5,5))
//ISPCTL1 DD DISP=(NEW,DELETE,DELETE),UNIT=SYSALLDA,
//          LRECL=80,RECFM=FB,SPACE=(TRK,(5,5))
//ISPMLIB DD DISP=SHR,DSN=ISP.SISPMENU
//ISPPLIB DD DISP=SHR,DSN=ISP.SISPPENU
//ISPSLIB DD DISP=SHR,DSN=ISP.SISPSENU
//ISPTLIB DD DISP=SHR,DSN=ISP.SISPTENU
//*
//* DD used by TSO (IKJEFT01)
//SYSPROC DD DISP=SHR,DSN=#hlq.SFEKPROC          * CRAALLOC
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
%CRALLOC ISPSTART PGM(CRASERV) PARM(&CRAPRM1 &CRAPRM2 &STR(+))
&CRAPRM3 &STR(+)
&CRAPRM4 &STR(+)
&CRAPRM5 &STR(+)
&CRAPRM6 &STR(+)
&CRAPRM7 &STR(+)
&CRAPRM8 &STR(+)
)
$$
EXIT CODE(0)

```

図 25. CRASUB*: バッチ実行依頼を使用した CARMA の始動

注:

- SYSTSIN データを変更する場合は、ストリングの処理を簡素化するために、
PARM() ステートメントの右丸括弧 (“)”) の前に少なくとも 1 つの空白を必ず置いてください。
- CARMA TSO 環境をカスタマイズするために、独自の DD ステートメントとデータ・セット連結を追加し、TSO ログオン・プロシーチャーを模倣することができます。
- オプションで、CRASUB* CLIST 内の SET CRAPRM2=420 行のコメントを外して変更することにより、CARMA のタイムアウト値を変更できます。このタイムアウト値は、CARMA がクライアントからの次のコマンドを待つ秒数です。0 の値を設定すると、デフォルトのタイムアウト値になります。つまり、現行の 420 秒 (7 分) です。

- サーバーの終了時に、CARMA 始動プロセスの詳細は、rsecomm.log に示されます。rsecomm.log の詳細レベルの設定について詳しくは、135 ページの『(オプション) rsecomm.properties、RSE トレース』を参照してください。

CARMA VSAM データ・セット

CARMA サーバーは、3 つの VSAM データ・セットへの READ アクセス権限を必要とします。これらの VSAM データ・セットを作成し、取り込むためのサンプル・メンバーは、FEK.#CUST.JCL に置かれます。ただし、FEK.SFEKSAMP(FEKSETUP) ジョブをカスタマイズして実行依頼したときに別のロケーションを指定した場合は除きます。詳しくは、20 ページの『カスタマイズのセットアップ』を参照してください。

注:

- (カスタムの) RAM の定義を既存の VSAM 構成にマージする必要がある場合は、サンプル・ジョブ FEK.#CUST.JCL(CRA#UADD) を参照してください。このジョブは、変更する CARMA VSAM ファイルごとにカスタマイズし、実行依頼する必要があります。さまざまな CARMA VSAM ファイルで使用されるレコード構造の詳細については、「*Common Access Repository Manager Developer's Guide*」(SC23-7660) を参照してください。
- VSAM から順次データ・セットへアクティブな定義を抽出するには、FEK.#CUST.JCL(CRA#UQRY) サンプル・ジョブを使用します。

CRADEF、構成データ・セット

VSAM データ・セットは、定義された RAM がサポートする機能を記述します。RAM 開発者はこのデータ・セットへの UPDATE アクセス権限が必要です。データ・セットは、以下のいずれかのサンプル・ジョブにより作成できます。

- CRA\$VCAD は CA Endeavor® SCM RAM データをデータ・セットに取り込みます。
- CRA\$VDEF はサンプル RAM データをデータ・セットに取り込みます。

前述のサンプル・ジョブを使用して、定義された RAM を VSAM 作成中に使用不可にすることができます。これを行うことによって、単一のマスター入力ファイル (IBM によって提供されたもの、または RAM 開発者によってカスタマイズされたもの) を使用して、カスタマイズされた CARMA セットアップを作成できます。

CRAMSG、メッセージ・データ・セット

この VSAM データ・セットは、CARMA サーバー自体が発行したメッセージを保持します。データ・セットは、以下のいずれかのサンプル・ジョブにより作成できます。

- CRA\$VMSG は汎用サーバー・データをデータ・セットに取り込みます。

CRASTRS、カスタム・ストリング・データ・セット

この VSAM データ・セットは、定義された RAM が発行したメッセージを保持します。RAM 開発者はこのデータ・セットへの UPDATE アクセス権限が必要です。データ・セットは、以下のいずれかのサンプル・ジョブにより作成できます。

- CRA\$VCAS は CA Endeavor® SCM RAM データをデータ・セットに取り込みます。
- CRA\$VSTR はサンプル RAM データをデータ・セットに取り込みます。

CARMA VSAM のマイグレーションに関する注

1. バージョン 7.6.1 以降、Developer for System z は、メッセージ長の制限を解消するために、CARMA カスタム情報 VSAM データ・セット CRASTRS の新しいデータ構造レイアウトをサポートしています。

Developer for System z バージョン 7.6.1 までは、CARMA カスタム情報 VSAM データ・セットで定義されるストリングが、事前に定義された長さに制限されます。この制限のために、RAM 開発者は記述ストリングを短くするか、フルサイズのストリングを表示するためにクライアント・サイド・プラグインを使用する必要があります。

新規の VSAM レコード構造は、CARMA カスタム情報 VSAM データ・セット CRASTRS の可変長データ構造レイアウトをサポートしています。このレイアウトでは、ストリングが固定長とならずに区切り文字で分離されます。

既存の固定長の CARMA カスタム情報 VSAM データ・セット CRASTRS を新しい可変長フォーマットに変換するには、FEK.SFEKSAMP(CRA#VS2) JCL をカスタマイズして実行依頼します。

注:

- バージョン 7.6.1 より、サンプルの CARMA カスタム情報 VSAM データ・セットは可変長フォーマットで提供されます。
- バージョン 7.6.1 より、CARMA ロード・モジュール CRASERV は、CARMA カスタム情報 VSAM データ・セットの固定長フォーマットと可変長フォーマットの両方をサポートします。
- 古いバージョンの CARMA ロード・モジュールは、可変長フォーマットをサポートしないため、可変長の CARMA カスタム情報 VSAM データ・セットとともに使用すると、ストリングの文字化けが発生します。

2. バージョン 8.5.1 より、CRADEF VSAM データ・セットに定義されている RAM を無効にできるようになりました。詳しくは、94 ページの『CRADEF、構成データ・セット』に記載されているサンプル CRADEF 作成ジョブ内のドキュメンテーションを参照してください。

CARMA Repository Access Manager (RAM)

Repository Access Manager (RAM) は、z/OS ベースの Software Configuration Manager (SCM) 用のアプリケーション・プログラミング・インターフェース (API) です。Developer for System z またはユーザー作成アプリケーションは CARMA サーバーを始動でき、CARMA サーバーは RAM をロードし、サポートされる SCM があればそれにアクセスする標準インターフェースを提供します。

CARMA サーバーは、LINKLIST または STEPLIB/TASKLIB を介して、RAM ロード・モジュールを見つけることができなければなりません。

Developer for System z により提供される CRAR* RAM ロード・モジュールは FEK.SFEKLOAD に置かれ、サンプルのソース・コードおよびコンパイル・ジョブは FEK.SFEKSAMP に置かれます。ただし、Developer for System z の SMP/E インストール時に別の高位修飾子を使用した場合は除きます。

以下のセクションには、Developer for System z で使用できる RAM に関するカスタマイズ上の注意があります。参照するサンプルのメンバーは FEK.#CUST.* に置かれます。ただし、FEK.SFEKSAMP(FEKSETUP) サンプル・ジョブをカスタマイズして実行依頼したときに、別のロケーションを指定した場合は除きます。詳しくは、20 ページの『カスタマイズのセットアップ』を参照してください。

CARMA に関する詳細な情報、および提供されているサンプル RAM およびサンプル・ソース・コードの詳細については、「*Common Access Repository Manager Developer's Guide*」(SC23-7660) を参照してください。

CA Endeavor® SCM RAM

- CA Endeavor® SCM RAM は実動タイプの RAM です。
- CA Endeavor® SCM RAM は、Developer for System z クライアントが CA Endeavor® SCM エlementに直接アクセスできるようにします。
- ロード・モジュール名は CRARNVDR です。
- CA Endeavor® SCM RAM には、標準的な CARMA のセットアップと比べて多数の追加設定があります。CA Endeavor® SCM RAM をサポートする事前構成済みのセットアップのいずれかを開始点として使用し、ニーズに合わせてカスタマイズしてください。
- 非推奨 TSO/ISPF クライアント・ゲートウェイ始動方式を、CA Endeavor® SCM RAM と併用することはできません。
- CA Endeavor® SCM RAM には、カスタマイズが可能な複数の構成ファイルがあります。詳しくは、98 ページの『CRACFG、CRASHOW、および CRATMAP、CA Endeavor® SCM RAM 構成ファイル』を参照してください。
- CA Endeavor® SCM RAM には割り振り exec FEK.SFEKPROC(CRANDVRA) があり、これもカスタマイズできます。詳しくは、99 ページの『CRANDVRA、CA Endeavor® SCM RAM 割り振り exec』を参照してください。
- CA Endeavor® SCM RAM は、(バックグラウンドにおける) バッチ・モードでの CA Endeavor® SCM アクションの実行をサポートしています。詳しくは、100 ページの『CA Endeavor® SCM RAM バッチ・アクション』を参照してください。

CA Endeavor® SCM パッケージ RAM

- CA Endeavor® SCM パッケージ RAM は、実動タイプの RAM です。
- CA Endeavor® SCM パッケージ RAM は、Developer for System z クライアントが CA Endeavor® SCM パッケージに直接アクセスできるようにします。
- ロード・モジュール名は CRARPKGS です。
- CA Endeavor® SCM パッケージ RAM にはカスタマイズ可能な設定がなく、CA Endeavor® SCM RAM と組み合わせて使用する必要があります。

PDS RAM

- PDS RAM はサンプル RAM です。実稼働環境では使用しないでください。
- PDS RAM は、「リモート・システム」ビューの「MVS ファイル」->「ユーザー・データ・セット」と同様のデータ・セット・リストを提供します。
- ロード・モジュール名は CRARPDS です。
- PDS RAM を使用するには、ISPF サービスが使用可能である必要があります。

- PDS RAM は、追加の VSAM データ・セットが DD CRARAM1 に割り振られている必要があります。この VSAM データ・セットは、FEK.#CUST.JCL(CRA#VPDS) サンプル・ジョブに割り振って、このジョブを準備することができます。カスタマイズの手順については、メンバー内のドキュメンテーションを参照してください。
- ソース・コードおよびコンパイル・ジョブは FEK.SFEKSAMP にあります。詳細については、「*Common Access Repository Manager Developer's Guide*」(SC23-7660)を参照してください。

スケルトン RAM

- スケルトン RAM はサンプル RAM です。実稼働環境では使用しないでください。
- スケルトン RAM は、ユーザーが独自の RAM を開発する際の開始点として使用できる機能フレームワークを提供します。
- ロード・モジュール名は CRARTEST です。
- ソース・コードおよびコンパイル・ジョブは FEK.SFEKSAMP にあります。詳細については、「*Common Access Repository Manager Developer's Guide*」(SC23-7660)を参照してください。

SCLM RAM

- SCLM RAM はサンプル RAM です。実稼働環境では使用しないでください。
- SCLM RAM は、ISPF の Software Configuration Manager である SCLM への基本的な入り口となります。この RAM は、デフォルトでは使用可能に設定されていません。
- ロード・モジュール名は CRARSCLM です。
- SCLM RAM を使用するには、ISPF サービスが使用可能である必要があります。
- SCLM RAM は、追加の VSAM データ・セットが DD CRARAM2 に割り振られている必要があります。この VSAM データ・セットは、FEK.#CUST.JCL(CRA#VSLM) サンプル・ジョブに割り振って、このジョブを準備することができます。カスタマイズの手順については、メンバー内のドキュメンテーションを参照してください。
- SCLM RAM を使用するには、さまざまなユーザー固有のデータ・セットが存在している必要があります。FEK.#CUST.JCL(CRA#ASLM) をカスタマイズして、これらのデータ・セットを割り振ります。カスタマイズの手順については、メンバー内のドキュメンテーションを参照してください。各ユーザーは、CARMA を SCLM RAM と一緒に使用する前に、1 回だけ CRA#ASLM を実行依頼する必要があります。そうしなかった場合は、割り振りエラーになります。
- SCLM RAM は、デフォルトでは使用可能に設定されていません。RAM を使用可能にするには、これを、DD CRADEF と CRASTRS が参照する CARMA VSAM データ・セットに定義する必要があります。FEK.#CUST.JCL(CRA#UADD) サンプル・ジョブを使用して、FEK.SFEKVSM2(CRA0SLMD) を CRADEF へ、FEK.SFEKVSM2(CRA0SLMS) を CRASTRS へマージします。カスタマイズの手順については、メンバー内のドキュメンテーションを参照してください。
- ソース・コードおよびコンパイル・ジョブは FEK.SFEKSAMP にあります。詳細については、「*Common Access Repository Manager Developer's Guide*」(SC23-7660)を参照してください。

CRACFG、CRASHOW、および CRATMAP、CA Endeavor® SCM RAM 構成ファイル

以下の CA Endeavor® SCM RAM に固有の CARMA コンポーネントは、選択したサーバー始動方式に関係なく、カスタマイズすることができます。参照するサンプルのメンバーは、FEK.#CUST.PARMLIB に置かれます。ただし、FEK.SFEKSAMP(FEKSETUP) ジョブをカスタマイズして実行依頼したときに、別のロケーションを指定した場合は除きます。詳しくは、20 ページの『カスタマイズのセットアップ』を参照してください。

CRACFG、CA Endeavor® SCM RAM と SCM の相互作用

CRACFG は、CA Endeavor® SCM RAM が CA Endeavor® SCM と相互作用する方法を指定します。デフォルトを変更する場合、カスタマイズの手順については、メンバー内のドキュメンテーションを参照してください。

```
# ENTRY-STAGE-COPY-MODE = RETRIEVE-ADD
# ALTERNATIVE-ALLOC = YES
```

図 26. CRACFG - CA Endeavor® SCM RAM と SCM の相互作用

CRASHOW、CA Endeavor® SCM RAM のデフォルト・フィルター

CRASHOW は、CA Endeavor® SCM の環境やシステムなどに対するデフォルト・フィルターを定義します。デフォルトを変更する場合、カスタマイズの手順については、メンバー内のドキュメンテーションを参照してください。

```
ENV=*
TOENV=
STGID=*
TOSTGID=
SYS=*
SUBSYS=*
ELEM=*
TOELEM=
TYPE=*
#FILTER-DEP=YES
```

図 27. CRASHOW - CA Endeavor® SCM RAM のデフォルト・フィルター

注: FILTER-DEP は、共通の CA Endeavor® SCM 変数ではなく、他の CA Endeavor® SCM リポジトリ・ロケーションへのフットプリントの参照によってエレメントの依存関係のスキャンを制御する Developer for System z 固有の変数です。

CRATMAP、CA Endeavor® SCM RAM のファイル拡張子のマッピング

CRATMAP は、CA Endeavor® SCM タイプとファイル拡張子のマッピングをオーバーライドします。デフォルトを変更する必要がある場合は、メンバー内のドキュメンテーションにある、カスタマイズに関する説明を参照してください。

```
# *      = cbl
# COBOL  = cbl
# COPY   = cpy
# ASM    = asm
# MACRO   = asm
# PROCESS = jcl
```

図 28. CRATMAP: CA Endeavor® SCM RAM のデフォルト・フィルター

CRANDVRA、CA Endeavor® SCM RAM 割り振り exec

バッチ実行依頼方式および CRASTART 始動方式はどちらも、CRANDVRA REXX exec を呼び出して、CA Endeavor® SCM RAM で使用するユーザー固有のデータ・セットを割り振ります。割り振りは別個の exec で行われます。これは、exec を使用した方が、バッチ実行依頼 CRASUBCA JCL や CRASTART crastart.endeavor.conf 構成ファイル内部で行う場合より高い柔軟性が得られるためです。また、割り振り exec はオプションのユーザー出口の呼び出しも行います。

DD	データ・セット名	タイプ
DEPEND	&SYSPREF..&SYSUID.. &SYSNAME..CRA\$NDVR.DEPEND	永続的
BROWSE	&SYSPREF..&SYSUID.. &SYSNAME..CRA\$NDVR.BROWSE	一時的
CIPRINT	&SYSPREF..&SYSUID.. &SYSNAME..CRA\$NDVR.LISTING	一時的
SPCLLIST	&SYSPREF..&SYSUID.. &SYSNAME..CRA\$NDVR.SPCLLIST	一時的
PKGSCLS	&SYSPREF..&SYSUID.. &SYSNAME..CRA\$NDVR.PKGSCLS	一時的
CRABJCLO	&SYSPREF..&SYSUID.. &SYSNAME..CRA\$NDVR.CRABJCLO	一時的
ENHCEDIT	&SYSPREF..&SYSUID.. &SYSNAME..CRA\$NDVR.ENHCEDIT	一時的
CRAPARM	&SYSPREF..&SYSUID.. &SYSNAME..CRA\$NDVR.CRAPARM	一時的

特定のデフォルト (データ・セット名など) がご使用のサイトの標準に適合しない場合は、この割り振り REXX exec のコピーをカスタマイズできます。CRANDVRA は FEK.SFEKPROC に置かれます。ただし、Developer for System z の SMP/E インストール時に別の高位修飾子を使用した場合は除きます。

カスタマイズの手順については、メンバー内のドキュメンテーションを参照してください。割り振り exec の詳細については、106 ページの『(オプション) カスタム割り振り exec』を参照してください。

注: サンプルの割り振り REXX を新しいデータ・セットにコピーし、そのコピーをカスタマイズして、保守の適用時に上書きされないようにしてください。これを行う場合は、選択した CARMA 始動方式の SYSEXEC DD で SFEKPROC への参照を更新してご使用の新しいデータ・セット名に合わせる必要があります。

CA Endeavor® SCM RAM バッチ・アクション

「エレメントの生成」などの CA Endeavor® SCM アクションは通常、CARMA サーバー・アドレス・スペースで「オンライン」で実行されます。使用する CA Endeavor® SCM プロシーチャーで TSO を呼び出す場合、この動作により問題が発生します。理由は、TSO が既にアクティブである、つまり、SYSTSIN や SYSTSPRT などの必要な DD が使用されているからです。

この問題を解決するために、CA Endeavor® SCM RAM はバージョン 8.0.3 から「バッチ・アクション」をサポートしています。バッチ・アクションを使用可能にすると、CA Endeavor® SCM RAM は、「エレメントの生成」などのアクションを実行するために、カスタマイズ可能なバッチ・ジョブを実行依頼します。バッチ・ジョブを使用した場合、CA Endeavor® SCM プロシーチャーによる SYSTSIN や SYSTSPRT などの DD の割り振りが行われます。これは、実行依頼された JCL では TSO がアクティブである必要はないためです。

CA Endeavor® SCM RAM バッチ・アクションは、バックグラウンドで実行する CA Endeavor® SCM アクションの Developer for System z 版です。

バッチ・アクションでサポートされるアクションを実行するように要求が出されると、CA Endeavor® SCM RAM は CRASUBCA または crastart.endeavor.conf で CRABCFG DD の有無を検査し、この DD のためのセットアップが有効か検査します。CRABCFG が存在し、そのセットアップが有効である場合、アクションはバッチで実行されます。CRABCFG が存在しない場合、アクションはオンラインで実行されます。バージョン 8.0.3 以降のクライアントは、この動作をオーバーライドする機能を備えています。

次に例を示します。

```
/* uncomment CRABCFG and CRABSKEL to use batch actions
/*CRABCFG DD DISP=SHR,DSN=FEK.#CUST.PARMLIB(CRABCFG)
/*CRABSKEL DD DISP=SHR,DSN=FEK.#CUST.CNTL
```

注:

- TSO が含まれない環境は、選択された CA Endeavor® SCM アクションでのみ使用可能です。バッチ・アクションは、この範囲外の TSO が含まれない環境をサポートしません。
- CRABCFG 構成ファイルは、サポートされている CA Endeavor® SCM アクションについて説明します。
- 機能サンプル・ジョブ FEK.#CUST.CNTL(CRABATCA) は、バッチ・アクションを実行するために用意されていますが、それらのバッチ・アクションの目的はこのサンプルが現行の CA Endeavor® SCM プロシーチャーを開始するようにカスタマイズされていることを示すことです。
- バッチ・アクション JCL の実行依頼に使用されるクラスに、十分な JES イニシエーターがあることを確認してください。
- SYSPLEX 環境で JES を使用する場合、ジョブが現在のシステムで稼働すること、または完了の情報が Developer for System z をホスティングしているシステムに送られることを確認します。これで、CA Endeavor® SCM RAM は状況をチェックできます。

- Developer for System z のクライアントとホスト・システムがバージョン8.5.1 以降の場合、クライアントは実行依頼をする前に、バッチ・アクション JCL に対して、カスタマイズ済みの JOB カードおよび追加の JCL ステートメントを提供することができます。

CRABCFG、CA Endeavor® SCM RAM バッチ・アクション構成

CRABCFG は、CA Endeavor® SCM RAM のバッチ・アクションに関連した構成変数を定義します。

CRABCFG は FEK.#CUST.PARMLIB に置かれます。ただし、FEK.SFEKSAMP(FEKSETUP) ジョブをカスタマイズして実行依頼したときに、別のロケーションを指定した場合は除きます。詳しくは、20 ページの『カスタマイズのセットアップ』を参照してください。

以下の CRABCFG サンプル・ファイルを参照してください。このファイルは、使用するシステム環境に合わせてカスタマイズする必要があります。US コード・ページを使用する場合、コメント行は番号記号 (#) で始まります。ディレクティブとそれに割り当てられる値の後のコメントは、サポートされています。等号 (=) の前後のスペースはサポートされています。行の継続はサポートされていません。

注: 変更は、更新後に始動されたすべての CARMA サーバーについて有効になります。

```
# Location of batch action JCL
SKELETON-DD = CRABSKEL
#
# batch action JCL members within SKELETON-DD
DEFAULT-JOBCARD = CRABJOBC
ADD-ELEMENT     = CRABATCA
DELETE-ELEMENT  = CRABATCA
GENERATE-ELEMENT = CRABATCA
MOVE-ELEMENT    = CRABATCA
RETRIEVE-ELEMENT = CRABATCA
PRINT-ELEMENT   = CRABATCA
PRINT-MEMBER    = CRABATCA
SIGNIN-ELEMENT  = CRABATCA
TRANSFER-ELEMENT = CRABATCA
#
# Command substitution key within batch action JCL
BSTIPT01-KEY = <CRA_BSTIPT01>
```

図 29. CRABCFG: CA Endeavor® SCM RAM バッチ・アクション構成

SKELETON-DD

バッチ・アクション・スケルトン JCL を保持する 1 つ以上の PDS(E) データ・セットを参照する DD ステートメントの名前。サンプル値は CRABSKEL です。必要であれば変更できます。この DD は、CRASUBCA または crastart.endeavor.conf で CARMA サーバーに対して定義する必要があります。

DEFAULT-JOBCARD

デフォルト JOB カードを持っているメンバーの名前。ユーザー固有の JOB カード (バージョン 8.5.1 以降の Developer for System z クライアントに格

納されている) によって却下されない限り、スケルトン JCL では、
<JOB CARD> キーはこのデフォルト JOB カードに置き換えられます。必要であれば変更できます。

GENERATE-ELEMENT およびその他の CA Endeavor® SCM アクション

キー名は、バッチ・アクションによってサポートされ、変更することができない CA Endeavor® SCM アクションを表します。それぞれのキーに割り当てられる値は、関連するスケルトン JCL のメンバー名です。すべてのキーのサンプル値は、CRABATCA です。必要であれば変更できます。

BSTIPT01-KEY

実際の CA Endeavor® SCM コマンド・ストリングの置換キー。サンプル値は <CRA_BSTIPT01> です。必要であれば変更できます。スケルトン JCL 内におけるこの置換キーの最初のオカレンス (コメントに含まれるものを除く) が、要求されたアクションを要求された要素に対して実行するように CA Endeavor® SCM に指示するコマンド・ストリングに置換されます。

CRABATCA、CA Endeavor® SCM RAM バッチ・アクション JCL

CRABATCA は、バッチ・アクションに使用されるサンプルのスケルトン JCL です。デフォルトを変更する場合は、メンバー内のドキュメンテーションでカスタマイズ手順を確認してください。

CRABATCA は FEK.#CUST.CNTL に置かれます。ただし、FEK.SFEKSAMP(FEKSETUP) ジョブをカスタマイズして実行依頼したときに、別のロケーションを指定した場合は除きます。詳しくは、20 ページの『カスタマイズのセットアップ』を参照してください。

変更は、すべての新規呼び出しについてアクティブになります。サーバーの再始動は必要ありません。

```
//<JOB CARD>
//*
//CRABATCA EXEC PGM=NDVRC1,DYNAMNBR=1500,REGION=4096K,PARM='C1BM3000'
//STEPLIB DD DISP=SHR,DSN=CA.NDVR.CSIQAUTU
// DD DISP=SHR,DSN=CA.NDVR.CSIQAUTH
//CONLIB DD DISP=SHR,DSN=CA.NDVR.CSIQLOAD
//C1MSG1 DD SYSOUT=*
//C1MSG2 DD SYSOUT=*
//C1PRINT DD SYSOUT=*,DCB=(RECFM=FBA,LRECL=133)
//SYSOUT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYMDUMP DD DUMMY
//SYSIN DD DUMMY
//BSTIPT01 DD *
SET STOPRC 16 .
<CRA_BSTIPT01>
/*
```

図 30. CRABATCA: CA Endeavor® SCM RAM バッチ・アクション JCL

CRABJOBC、CA Endeavor® SCM RAM バッチ・アクション JOB カード

CRABJOBC は、<JOBCARD> キーを指定するバッチ・アクション・スケルトン JCL に使用されるサンプルのデフォルト JOB カードです。デフォルトを変更する場合は、メンバー内のドキュメンテーションでカスタマイズ手順を確認してください。

CRABJOBC は FEK.#CUST.CNTL に置かれます。ただし、FEK.SFEKSAMP(FEKSETUP) ジョブをカスタマイズして実行依頼したときに、別のロケーションを指定した場合は除きます。詳しくは、20 ページの『カスタマイズのセットアップ』を参照してください。

変更は、すべての新規呼び出しについてアクティブになります。サーバーの再始動は必要ありません。

```
//<USERID>B JOB CLASS=A,MSGCLASS=A,MSGLEVEL=(1,1)
//*PROCS JCLLIB ORDER=(COBOL.V4R1M0.SIGYPROC,CBC.SCCNPRC)
```

図 31. CRABJOBC: CA Endeavor® SCM RAM バッチ・アクション JOB カード

CRAALLOC、カスタム RAM 割り振り exec

バッチ実行依頼方式および CRASTART 始動方式はどちらも、CRAALLOC REXX exec を呼び出して、ユーザー作成 RAM で使用できるユーザー固有のデータ・セットを割り振ります。割り振りは別個の exec で行われます。これは、exec を使用した方が、バッチ実行依頼 CRASUBMT JCL や CRASTART crastart.conf 構成ファイル内部で行う場合より高い柔軟性が得られるためです。また、割り振り exec はオプションのユーザー出口の呼び出しも行います。

DD	データ・セット名	タイプ
CRAPARM	&SYSPREF..&SYSUID.. &SYSNAME..CRA\$CUST.CRAPARM	一時的

特定のデフォルト (データ・セット名など) がご使用のサイトの標準に適合しない場合は、この割り振り REXX exec のコピーをカスタマイズできます。CRAALLOC は FEK.SFEKPROC に置かれます。ただし、Developer for System z の SMP/E インストール時に別の高位修飾子を使用した場合は除きます。

カスタマイズの手順については、メンバー内のドキュメンテーションを参照してください。割り振り exec の詳細については、(オプション) カスタム割り振り exec を参照してください。

注: サンプルの割り振り REXX を新しいデータ・セットにコピーし、そのコピーをカスタマイズして、保守の適用時に上書きされないようにしてください。これを行う場合は、選択した CARMA 始動方式の SYSEXEC DD で SFEKPROC への参照を更新してご使用の新しいデータ・セット名に合わせる必要があります。

CARMA 戻りコード

CARMA は、さまざまなエラー・コードをクライアントに対して、またはホスト・システム・ログで報告できます。エラーとともに提供される詳細と、表 13 の情報は、エラーを見つけて解決に取り組むのに役立ちます。

表 13. CARMA 戻りコード

エラー範囲	エラー・タイプ
4 から 99	一般的な CARMA エラー
100 から 199	一般的な RAM エラー
200 から 399	CRASERV (CARMA サーバー) エラー
400 から 499	RSE (CARMA マイナー) エラー
500 から 899	RAM 固有のエラー
900 から 999	TSO エラーおよび TCP/IP エラー

いくつかの一般的な戻りコードを以下に示します。

- 220: 非アクティブ・タイムアウトのため CARMA サーバーが終了します。これはエラーではありません。
- 990: CARMA マイナーが listen しているポートに CARMA サーバーが接続できません。

(オプション) 複数の RAM のサポート

CARMA は、複数の RAM を定義して同時に実行するための機能を備えています。ただし、1 ユーザーにつきアクティブな CARMA サーバーは 1 つのみであるため、複数の RAM がある場合でも、そのセットアップを機能させるために構成の変更が必要となることがあります。

RAM は、RAM 開発者によって CARMA 構成 VSAM データ・セット CRADEF 内で定義されます。CARMA サーバー CRASERV は、始動時に、定義されているすべての RAM を識別し、その情報を CARMA クライアントに送信します。これでユーザーは 1 つ以上の RAM を選択できるようになり、それらが CARMA サーバーにロードされます。

RAM は CARMA サーバーのプラグインとしてアクティブになるため、各 RAM の前提条件となるもの (データ・セット割り振りなど) がすべて CARMA サーバーのアドレス・スペースにあることを確認してください。この要件のために、Developer for System z に付属する CARMA 構成サンプル (CRASUBMT や crastart.conf など) の変更が必要になることがあります。

例

以下の例では、CA Endevor® SCM RAM を使用する既存のセットアップから CRASTART 始動方式を使用して始動し、サンプルの PDS RAM を追加します。

CA Endevor® SCM RAM 用の定義:

- FEK.SFEKVSM2(CRA0VCAD): CRADEF 定義
- FEK.SFEKVSM2(CRA0VCAS): CRASTRS 定義

- /etc/rdz/crastart.endevor.conf: CRASTART 構成ファイル

PDS RAM 用の定義:

- FEK.SFEKVSM2(CRA0VDEF): CRADEF 定義
- FEK.SFEKVSM2(CRA0VSTR): CRASTRS 定義
- FEK.#CUST.CRARAM1: CRARAM1 定義

このプロセスは、システム・プログラマーがセットアップを完了するために必要とするデータと情報を、RAM 開発者が収集することから始まります。

1. PDS RAM に固有のデータを SFEKVSM2 メンバーから抽出します。これらのメンバーは、PDS RAM だけでなくすべてのサンプル RAM の定義を保持しています。
2. このデータを CA Endeavor® SCM RAM SFEKVSM2 メンバーとマージします。
3. PDS RAM 固有の前提条件のリストを作成します。
 - FEK.#CUST.CRARAM1 にリンクされた DD CRARAM1
 - TSO 環境

次に、システム・プログラマーがこのデータを使用して更新された CARMA VSAM データ・セットを作成し、前提条件情報を使用して、両方の RAM をサポートできる CRASTART 構成ファイルを作成します。

1. この結合データを、CRA\$VDEF および CRA\$VSTR ジョブへの入力として使用して、更新された CARMA 構成および CARMA カスタム情報 VSAM データ・セット (CRADEF と CRASTRS) を作成します。CRAMSG VSAM は CARMA サーバーに固有のものであり、したがって両方の RAM で同一です。
2. CRARAM1 定義を crastart.endevor.conf に追加します。
CRARAM1 = FEK.#CUST.CRARAM1
3. crastart.endevor.conf 内の PROGRAM ステートメントを検証して、両方の RAM に必要な環境を提供できることを確認します。
4. PROGRAM=IKJEFT01 %CRANDVRA NDVRC1 PGM(CRASERV)
PARM(&CRAPRM1. &CRAPRM2.)
 - IKJEFT01: 無許可の環境で特定の許可された呼び出しを可能にするために使用され、また、CA Endeavor® SCM RAM 事前割り振り exec を実行する環境として使用される TSO。
 - %CRANDVRA: 一時的なユーザー固有の作業データ・セットを割り振る CA Endeavor® SCM RAM 事前割り振り exec (FEK.SFEKPROC に配置)。
 - NDVRC1: TSO コマンドと ISPF コマンドを実行するメカニズムが組み込まれた、CA Endeavor® バックエンド。
 - PGM(CRASERV): CARMA サーバーを始動する、ISPF コマンド・フォーマットのコマンド。
 - PARM(&CRAPRM1. &CRAPRM2.): ISPF コマンド・フォーマットの CRASERV のパラメーター。&CRAPRM1 は使用されるポート、&CRAPRM2 はタイムアウト値です。

CA Endeavor® SCM RAM は ISPF 環境でアクティブになります。これは、PDS RAM に必要な TSO 環境も使用可能であることを意味します。

(オプション) カスタム割り振り exec

すべての CARMA サーバー始動方式には、データ・セットの割り振りに関する制約があります。例えば、TSO 接頭辞の置換は、JCL や CRASTART では使用できません。

ただし、TSO または ISPF の始動後、かつ CARMA の始動前に呼び出される exec を作成すると、TSO または ISPF で使用可能なすべての範囲の変数とサービスを使用して必要な割り振りを実行することが可能になります。

Developer for System z は、この章で先に説明した各事前構成済みセットアップで割り振り exec を使用します。FEK.SFEKPROC(CRANDVRA) (CA Endeavor® SCM RAM の割り振り exec) および FEK.SFEKPROC(CRAALLOC) (カスタム RAM の割り振り exec) であり、exec は、ユーザーの TSO 接頭部を高位修飾子に持つ、カタログされた一時データ・セットを割り振ります。また、割り振り exec はオプションのユーザー出口の呼び出しも行います。

カスタマイズについての説明は、exec 内に記載されています。割り振り exec の変更はサポートされていますが、推奨されません。それは、PTF サービスによって exec が更新された場合に、カスタマイズを再度行う必要があるためです。可能であれば、代わりに CARMA ユーザー出口 (107 ページの『(オプション) CARMA ユーザー出口』を参照) を使用してください。

注:

- 割り振り exec を更新するときは、CARMA 始動プロセスで既に行われた割り振りを、CRASTART または始動 JCL によって破壊することがないようにしてください。
- 割り振り exec により生成される出力は、CARMA サーバーの DD SYSTSPRT に示されます。

割り振り exec を更新するときは、CARMA 始動プロセスで既に行われた割り振りを、CRASTART または始動 JCL によって破壊することがないようにしてください。

以下のサンプルは、TSO のみを必要とする割り振り exec の始動方法を示しています。

```
crastart*.conf
```

```
SYSPROC = my.exec.library  
PROGRAM = IKJEFT01 %myexec ISPSTART PGM(CRASERV) PARM(&CRAPRM1. &CRAPRM2. )
```

```
CRASUB*
```

```
//SYSPROC DD DISP=SHR,DSN=my.exec.library  
//SYSTSIN DD *  
%myexec ISPSTART PGM(CRASERV) PARM(&CRAPRM1. &CRAPRM2. )  
//*
```

(オプション) CARMA ユーザー出口

CARMA はユーザー出口の呼び出しをサポートしており、CARMA サーバーの始動時に特殊な初期化を行ったり、シャットダウン時に特殊なクリーンアップを行ったりすることができます。ユーザー出口を使用することで、PTF サービスによって保守されている割り振り exec の変更の必要性が減ります。

ユーザー出口は割り振り exec によって呼び出され、2 回実行されます。初期化の呼び出しは、一時データ・セットが割り振られてから CARMA サーバーが呼び出されるまでの間に行われます。クリーンアップの呼び出しは、CARMA サーバーが終了してから一時ファイルが除去されるまでの間に行われます。最初の呼び出しが戻りコード 99 以上で終了すると、CARMA 始動は中断します。これは、CARMA サーバーも、このユーザー出口の 2 回目の呼び出しも実行されないことを意味します。

FEK.SFEKSAMP(CRAEXIT) というサンプル・ユーザー出口が提供されます。ただし、Developer for System z の SMP/E インストール時に別の高位修飾子を使用した場合は除きます。このサンプル・ユーザー出口では、ユーザー出口に渡される以下の始動引数について詳しく説明しています。

始動引数	説明
(STARTUP) (ENDING)	出口呼び出しが CARMA サーバーの呼び出しの前であるか後であるかの標識。
EXIT_RC=rc	出口の前回の呼び出しの戻りコード。 (STARTUP) 呼び出しのときは rc は常に 0 です。
CARMA_RC=rc	CARMA サーバーの呼び出しの戻りコード。 (STARTUP) 呼び出しのときは rc は常に 0 です。
...	CARMA サーバーの始動コマンドと始動引数。例: ISPSTART PGM(CRASERV) PARM(1312 420 EXIT=CRAEXIT CLIENT=9.0.1 . . .)

ユーザー出口によって生成された出力は、CARMA サーバーの DD SYSTSPRT に表示されます。

(オプション) IRXJCL と CRAXJCL

TSO (IKJEFTxx) を使用して CARMA サーバーを始動する場合に、使用する RAM でサービスを呼び出し、さらにそのサービスで IRXJCL REXX バッチ・インターフェースを呼び出すと、問題が発生することがあります。問題が発生する可能性があるのは、RAM によって呼び出されたプロセッサが、以前は TSO なしで稼働していたかオンライン TSO 内でのみ稼働しており、それが DD SYSTSIN または SYSTSPRT を動的に割り振ったときです。この問題を回避するために、サンプル・プログラム CRAXJCL が提供されています。

プロセッサは、SYSTSIN または SYSTSPRT (IRXJCL に必須) を割り振ろうとした場合に失敗することがあります。これは、CARMA に必須のバッチ TSO が既に

それらの DD 名を割り振って開いているためです。CRAXJCL 置換モジュールは SYSTSIN および SYSTSPRT を DUMMY に割り振ろうとしますが、割り振りが失敗した場合に発生するエラーを無視します。その後、IRXJCL を呼び出して、実際の処理を行います。

このことは、TSO によって始動された CARMA 環境内でプロセッサを稼働した場合、SYSTSIN および SYSTSPRT への割り振りは CARMA によって使用される割り振りと同じものであることを意味します。TSO/CARMA の外部でプロセッサを稼働した場合、SYSTSIN および SYSTSPRT の割り振りは CRAXJCL によって作成されます。したがって、使用するプロセッサは、SYSTSIN へ割り振られたデータ・セットの内容に依存してはなりません。

IRXJCL の呼び出しは、「TSO/E REXX 解説書」(SA88-8635) で説明されているように、PARM フィールドを使用して REXX 名および始動パラメーターを渡すものと想定されています。これは、SYSTSIN が CARMA で安全に使用できることを意味します。IRXJCL によって SYSTSPRT へ送られたすべての出力は、CARMA のログに書き込まれます。

CRAXJCL 置換モジュールを呼び出すプロセッサは、CRAXJCL を呼び出す前に DD SYSTSIN または SYSTSPRT の割り振りを試みてはなりません。

CRAXJCL の作成

CRAXJCL 置換モジュールはソース形式で提供されています。これは、お客様がこのモジュールをカスタマイズして、SYSTSPRT に使用する固有の割り振りを指定する必要があるためです。SYSTSIN の割り振りは通常、ダミー・データ・セットに対して行う必要があります。

サンプルのアセンブラー・ソース・コードおよびサンプルのコンパイル/バインド・ジョブは、FEK.#CUST.ASM(CRAXJCL) および FEK.#CUST.JCL(CRA#CIRX) として使用可能です。ただし、FEK.SFEKSAMP(FEKSETUP) ジョブをカスタマイズして実行依頼したときに、別のロケーションを指定した場合は除きます。詳しくは、20 ページの『カスタマイズのセットアップ』を参照してください。

メンバー内のドキュメンテーションを使用し、必要に応じて CRAXJCL アセンブラー・ソース・コードをカスタマイズしてください。後で、CRA#CIRX JCL をカスタマイズおよび実行依頼して、CRAXJCL ロード・モジュールを作成します。カスタマイズの手順については、メンバー内のドキュメンテーションを参照してください。

必要に応じ、IRXJCL を別の名前に変更できます。IRXJCL の新規名を呼び出してコンパイルするように CRAXJCL ソースを調整してから、CRAXJCL ロード・モジュールの名前を IRXJCL に変更します。このセットアップは、IRXJCL へのすべての呼び出しを変更するよりも容易である可能性があります。

第 4 章 (オプション) SCLM Developer Toolkit

SCLM Developer Toolkit は、SCLM の機能を拡張するために必要なツールをクライアントに提供します。SCLM 自体はホスト・システム・ベースのソース・コード・マネージャーであり、ISPF の一部として組み込まれています。

SCLM Developer Toolkit は、SCLM へのインターフェースとして機能する Eclipse ベースのプラグインを備えています。これにより、従来型コード開発におけるすべての SCLM プロセスへのアクセスを可能にし、メインフレーム上の SCLM と同期したワークステーション上で完全な Java および Java EE 開発 (メインフレームからの Java EE コードのビルド、アセンブル、およびデプロイメントを含む) を行えるようにサポートします。

要件およびチェックリスト

このカスタマイズ・タスクを完了するには、SCLM 管理者、およびオプションとしてセキュリティー管理者の支援が必要になります。このタスクには、以下のリソースか特殊なカスタマイズ・タスクが必要です。

- APF および LINKLIST の更新
- Java EE サポート用の SCLM 言語変換プログラムを定義する
- Java EE サポート用の SCLM タイプを定義する
- (オプション) ユーザーに SCLM VSAM の更新を許可するセキュリティー規則
- (オプション) Ant のインストール

ご使用のサイトで SCLM Developer Toolkit の使用を開始するには、以下のタスクを行う必要があります。特に断りがない限り、すべてのタスクは必須です。

1. 前提条件と PARMLIB 更新を確認および調整します。詳細については、110 ページの『前提条件』を参照してください。
2. Developer for System z 構成ファイルをカスタマイズします。詳細については、以下を参照してください。
 - 110 ページの『SCLMDT 用の ISPF.conf の更新』
 - 111 ページの『SCLMDT 用の rsed.envvars の更新』
3. オプションとして、ロング/ショート・ネーム変換サポートを定義します。詳細については、112 ページの『(オプション) ロング/ショート・ネーム変換』を参照してください。
4. オプションとして、Java EE ビルド・サポートを使用するために Ant をインストールおよびカスタマイズします。詳細については、114 ページの『(オプション) Ant のインストールおよびカスタマイズ』を参照してください。
5. SCLM を更新して、SCLMDT 固有の部分を定義します。詳細については、115 ページの『SCLMDT 用の SCLM の更新』を参照してください。
6. オプションとして、SCLMDT 作業域の定期的な自動クリーンアップをセットアップします。詳細については、116 ページの『WORKAREA および /tmp からの古いファイルの除去』を参照してください。

前提条件

必要な SCLM メンテナンスのリストについては、「*IBM Rational Developer for System z 前提条件*」(SC88-4704) を参照してください。

この資料には、SCLM Developer Toolkit 内の Java EE ビルドに必要な Ant 仕様も記載されています。

重要: SCLM Developer Toolkit では ISPF の TSO/ISPF クライアント・ゲートウェイを使用します。これは、z/OS 1.8 以降が必要であることを意味します。

21 ページの『PARMLIB の変更』の説明にあるように、SCLM Developer Toolkit を使用するには、システム設定の追加のカスタマイズが必要です。これには以下の項目が含まれます。

- (BPXPRMxx) z/OS UNIX ユーザー ID の 1 つ当たりの最大プロセス数を増やします。
- (PROGxx) SYS1.LINKLIB および REXX ランタイム REXX.V1R4M0.SEAGLPA または REXX.V1R4M0.SEAGALT の APF 許可を与えます。
- (PROGxx/LPALSTxx) ISP.SISPLPA、ISP.SISPLPAD、SYS1.LINKLIB、および REXX ランタイムを LINKLIST/LPALIB に入れます。

また、SCLM Developer Toolkit は SDSF または TSO **OUTPUT** コマンドを使用して、ジョブの完了状況とジョブの出力を取り出します。どちらの方法でも、以下の点にさらに注意が必要です。

- SDSF は、別途オーダーし、インストールし、構成する必要があります。また、SDSF は JES2 を必要とします。
- TSO **OUTPUT** コマンドのデフォルトの設定では、各ユーザーは、自分の特定のユーザー ID で始まるジョブ出力だけを取り出すことができます。**OUTPUT** 機能を十分に使用するには、ユーザーが自分の所有するジョブ出力を、それが自分のユーザー ID で始まるものでなくても取り出すことができるように、サンプルの TSO/E 出口 IKJEFF53 を変更しなければならない場合があります。この出口について詳しくは、「*TSO/E カスタマイズ*」(SA88-8629) を参照してください。

ユーザーには、z/OS UNIX ディレクトリー /tmp/ および /var/rdz/WORKAREA/ に対する READ、WRITE、および EXECUTE 権限が必要です。ディレクトリー WORKAREA/ は /var/rdz/ に置かれます。ただし、FEK.SFEKSAMP(FEKSETUP) ジョブをカスタマイズして実行依頼したときに、別のロケーションを指定した場合は除きます。詳しくは、20 ページの『カスタマイズのセットアップ』を参照してください。

SCLMDT 用の ISPF.conf の更新

SCLM Developer Toolkit は標準の ISPF/SCLM スケルトンを使用するため、ISP.SISPSLIB スケルトン・ライブラリーが ISPF.conf 内の ISPSLIB 連結に割り振られるようにしてください。ISP.SISPSENU データ・セットの使用はオプションです。

ISPF.conf は /etc/rdz/ に置かれます。ただし、FEK.SFEKSAMP(FEKSETUP) ジョブをカスタマイズして実行依頼したときに、別のロケーションを指定した場合は除きます。詳しくは、20 ページの『カスタマイズのセットアップ』を参照してください。このファイルは、TSO **OEDIT** コマンドで編集できます。

注: 変更は、更新後にホスト・システムに接続するすべてのクライアントについて有効になります。

次のサンプル・コードは ISPF.conf ファイルを示しています。このファイルは、使用するシステム環境に合わせてカスタマイズする必要があります。コメント行は、アスタリスク (*) で始まります。同じ行にある連結にデータ・セットを追加し、名前同士をコンマ (,) で分離します。ISPF.conf のカスタマイズについて詳しくは、67 ページの『ISPF.conf、ISPF の TSO/ISPF クライアント・ゲートウェイ構成ファイル』を参照してください。

```
* REQUIRED:
sysproc=ISP.SISPLIB,FEK.SFEKPROC
isplib=ISP.SISPMENU
isptlib=ISP.SISPTENU
ispplib=ISP.SISPPENU
ispslib=ISP.SISPSLIB
ispllib=ISP.SISPLOAD
```

```
* OPTIONAL:
*allocjob = ISP.SISPSAMP(ISPZISP2)
*ISPF_timeout = 900
```

図 32. SCLMDT 用の ISPF.conf の更新

注:

- TSO 環境をカスタマイズするために、独自の DD のようなステートメントとデータ・セット連結を追加し、TSO ログオン・プロシーチャーを模倣することができます。詳しくは、「ホスト構成リファレンス」(SA88-4226) の『TSO 環境のカスタマイズ』を参照してください。
- バッチ・ビルドを行う場合は、ISPF/SCLM スケルトン・ライブラリーの前に、カスタマイズされたバージョンの FLMLIBS スケルトンが連結されるようにしてください。

```
ispslib=h1q.USERSKEL,ISP.SISPSLIB
```

SCLMDT 用の rsed.envvars の更新

SCLM Developer Toolkit は、rsed.envvars の中で設定されたいくつかのディレクティブを使用して、データ・セットおよびディレクトリーを見つけます。

rsed.envvars は /etc/rdz/ に置かれます。ただし、FEK.SFEKSAMP(FEKSETUP) ジョブをカスタマイズして実行依頼したときに、別のロケーションを指定した場合は除きます。詳しくは、20 ページの『カスタマイズのセットアップ』を参照してください。このファイルは、TSO **OEDIT** コマンドで編集できます。

注: 変更を有効にするには、RSED 開始タスクを再始動してください。

次のコード・サンプルは、rsed.envvars ファイル内の SCLMDT ディレクティブを示しています。このファイルは、使用するシステム環境に合わせてカスタマイズす

る必要があります。`rsed.envvars` のカスタマイズについて詳しくは、43 ページの『`rsed.envvars`、RSE 構成ファイル』を参照してください。

```
_SCLMDT_CONF_HOME=/var/rdz/sclmdt
#STEPLIB=$STEPLIB:FEK.SFEKAUTH:FEK.SFEKLOAD
#_SCLMDT_TRANTABLE=FEK.#CUST.LSTRANS.FILE
#ANT_HOME=/usr/lpp/apache/Ant/apache-ant-1.7.1
_SCLMDT_BASE_HOME=$RSE_HOME
_SCLMDT_WORK_HOME=$CGI_ISPHOME
CGI_DTWORk=$_SCLMDT_WORK_HOME
```

図 33. `SCLMDT` 用の `rsed.envvars` の更新

(オプション) ロング/ショート・ネーム変換

SCLM Developer Toolkit では、ロング・ネーム・ファイルを SCLM に保管することができます。ロング・ネーム・ファイルとは、8 文字を超える名前または大/小文字混合の名前を持つファイルのことです。ロング・ファイル・ネームの保管は、ロング・ファイル・ネーム名を、SCLM で使用する 8 文字のメンバー名に対応付けるマッピングが入った VSAM ファイルを使用することで実現しています。

注:

- z/OS 1.8 より前のバージョンの場合、この機能は APAR OA11426 に対応する基本 ISPF/SCLM PTF によって提供されます。
- ロング/ショート・ネーム変換は、IBM SCLM Administrator Toolkit などの、他の SCLM 関連製品でも使用されます。

LSTRANS.FILE (ロング/ショート・ネーム変換 VSAM) の作成

ロング/ショート・ネーム変換 VSAM を作成するには、ISP.SISPSAMP ISPF サンプル・ライブラリー内のサンプル・メンバー `FLM02LST` をカスタマイズして実行依頼します。本資料の構成ステップでは、以下のサンプルのセットアップ JCL に示すように、この VSAM に `FEK.#CUST.LSTRANS.FILE` という名前を付ける必要があります。

```

//FLM02LST JOB <job parameters>
/*
/* CAUTION: This is neither a JCL procedure nor a complete job.
/* Before using this sample, you will have to make the following
/* modifications:
/* 1. Change the job parameters to meet your system requirements.
/* 2. Change ***** to the volume that will hold the VSAM.
/* 3. Change all references of FEK.#CUST.LSTRANS.FILE to
/*     match your naming convention for the SCLM translate VSAM.
/*
//CREATE EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE FEK.#CUST.LSTRANS.FILE
SET MAXCC=0
DEFINE CLUSTER(NAME(FEK.#CUST.LSTRANS.FILE) -
               VOLUMES(*****)) -
               RECORDSIZE(58 2048) -
               SHAREOPTIONS(3 3) -
               CYLINDERS(1 1) -
               KEYS(8 0) -
               INDEXED) -
DATA (NAME(FEK.#CUST.LSTRANS.FILE.DATA)) -
INDEX (NAME(FEK.#CUST.LSTRANS.FILE.INDEX))

/* DEFINE ALTERNATE INDEX WITH NONUNIQUE KEYS -> ESDS */

DEFINE ALTERNATEINDEX(-
               NAME(FEK.#CUST.LSTRANS.FILE.AIX) -
               RELATE(FEK.#CUST.LSTRANS.FILE) -
               RECORDSIZE(58 2048) -
               VOLUMES(*****)) -
               CYLINDERS(1 1) -
               KEYS(50 8) -
               UPGRADE -
               NONUNIQUEKEY) -
DATA (NAME(FEK.#CUST.LSTRANS.FILE.AIX.DATA)) -
INDEX (NAME(FEK.#CUST.LSTRANS.FILE.AIX.INDEX))

/*
/*
//PRIME EXEC PGM=IDCAMS,COND=(0,LT)
//SYSPRINT DD SYSOUT=*
//INITREC DD *
INITREC1
/*
//SYSIN DD *
REPRO INFILE(INITREC) -
      OUTDATASET(FEK.#CUST.LSTRANS.FILE)
IF LASTCC = 4 THEN SET MAXCC=0

BLDINDEX IDS(FEK.#CUST.LSTRANS.FILE) -
          ODS(FEK.#CUST.LSTRANS.FILE.AIX)

IF LASTCC = 0 THEN -
  DEFINE PATH (NAME(FEK.#CUST.LSTRANS.FILE.PATH) -
              PATHENTRY (FEK.#CUST.LSTRANS.FILE.AIX))
/*

```

図 34. FLM02LST: ロング/ショート・ネーム変換セットアップ JCL

注: 「ホスト構成リファレンス」 (SA88-4226) の『セキュリティに関する考慮事項』の説明のように、ユーザーには、この VSAM データ・セットに対する UPDATE 権限が必要です。

ロング/ショート・ネーム変換用の rsed.envvars の更新

ロング/ショート・ネーム変換を使用する前に、rsed.envvars 環境変数 `_SCLMDT_TRANTABLE` をコメント解除し、ロング/ショート・ネーム変換 VSAM の名前に一致するように設定します。

rsed.envvars は /etc/rdz/ に置かれます。ただし、FEK.SFEKSAMP(FEKSETUP) ジョブをカスタマイズして実行依頼したときに、別のロケーションを指定した場合は除きます。詳しくは、20 ページの『カスタマイズのセットアップ』を参照してください。このファイルは、TSO **OEDIT** コマンドで編集できます。

注: 変更を有効にするには、RSED 開始タスクを再始動してください。

(オプション) Ant のインストールおよびカスタマイズ

このステップは、SCLM で Java EE ビルド・サポートを使用する場合にのみ、必要となります。

Apache Ant はオープン・ソースの Java ビルド・ツールであり、<http://ant.apache.org/> からダウンロードできます。Ant はテキスト・ファイルとスクリプトからなり、それらは ASCII フォーマットで配布されています。このため、z/OS UNIX で実行するためには、ASCII/EBCDIC 変換が必要です。

以下のステップを実行して、Ant を z/OS に実装し、Developer for System z に対して定義します。

- 最新の Ant 圧縮ファイルをバイナリー・フォーマットで z/OS UNIX ファイル・システムにダウンロードします。.zip 版の ANT をダウンロードします。
- z/OS UNIX コマンド行セッションを開き、TSO OMVS コマンドを使用するなどして、インストールを続行します。
- **mkdir -p /home-dir** コマンドを使用して Ant インストールのホーム・ディレクトリーを作成し、それを **cd /home-dir** コマンドで現行ディレクトリーにします。
- JAR 抽出コマンド **jar -xf apache-ant-1.7.1.zip** を使用して、ファイルを現行ディレクトリーに抽出してください。jar コマンドを使用するには、Java bin ディレクトリーがローカルの z/OS UNIX パスに存在していなければなりません。そうならない場合は、コマンドを Java bin ロケーションで完全修飾 (例えば、/usr/lpp/java/J6.0/bin/jar -xf apache-ant-1.7.1.zip) してください。
- すべての Ant テキスト・ファイルを EBCDIC に変換するため、オプションとして /usr/lpp/rdz/samples/BWBTRANT サンプル・スクリプトをカスタマイズして実行します。

注: このスクリプトは 1 回だけ実行してください。複数回実行すると、Ant インストールが壊れます。

- 変換が成功したかどうかを検査するために、ANT ディレクトリー内のテキスト・ファイル、例えば apache-ant-1.7.1/README などを見つけて開きます。ファイルが読み取り可能であれば、変換は正常に行われています。
- **chmod -R 755 *** コマンドを使用して、すべてのユーザーが ANT ディレクトリー内のファイルの読み取りと実行を行うことができるようにします。

- Ant を使用する前に、rsed.envvars の環境変数 JAVA_HOME および ANT_HOME を設定します。
 - JAVA_HOME は Java ホーム・ディレクトリーを指す必要があります。次に例を示します。
`JAVA_HOME=/usr/lpp/java/IBM/J6.0`
 - ANT_HOME は、次の例のように、Ant ホーム・ディレクトリーを指す必要があります。
`ANT_HOME=/usr/lpp/Apache/Ant/apache-ant-1.7.1`

次に例を示します。

- TSO OMVS
- `mkdir -p /usr/lpp/Apache/Ant`
- `cd /usr/lpp/Apache/Ant`
- `jar -xf /u/userid/apache-ant-1.7.1`
- `/usr/lpp/rdz/samples/BWBTRANT`
- `cat ./apache-ant-1.7.1/README`
- `chmod -R 755 *`
- `oedit /etc/rsed.envvars`

Ant の初期化が正常に行われたことをテストするには、次のようにします。

- Ant および Java bin ディレクトリーを環境変数 PATH に追加します。

例:

```
export PATH=/usr/lpp/Apache/Ant/apache-ant-1.7.1/bin:$PATH
export PATH=/usr/lpp/java/J6.0/bin:$PATH
```

- 正常にインストールが終了したら、バージョンを表示するために、`ant -version` を実行します。

例:

```
ant -version
```

注: この方法による PATH ステートメントの設定はテスト時に必要なものであって、実際の運用に際しては不要です。

SCLMDT 用の SCLM の更新

SCLM 自体も、SCLM Developer Toolkit を処理するためにはカスタマイズが必要です。以下の必要なカスタマイズ・タスクの詳細については、「*IBM Rational Developer for System z SCLM Developer Toolkit 管理者ガイド*」(SC88-5664) を参照してください。

- Java EE サポート用の言語変換プログラムを定義する
- Java EE サポート用の SCLM タイプを定義する

カスタマイズ・タスクおよびプロジェクト定義タスクを完了するためには、SCLM 管理者は、116 ページの表 14 に示す Developer for System z のカスタマイズ可能値を知っている必要があります。

表 14. SCLM 管理者チェックリスト

説明	<ul style="list-style-type: none"> デフォルト値 正解の入手先 	値
Developer for System z サンプル・ライブラリー	<ul style="list-style-type: none"> FEK.SFEKSAMV SMP/E インストール 	
Developer for System z サンプル・ディレクトリー	<ul style="list-style-type: none"> /usr/lpp/rdz/samples SMP/E インストール 	
Java bin ディレクトリー	<ul style="list-style-type: none"> /usr/lpp/java/J6.0/bin rsed.envvars - \$JAVA_HOME/bin 	
Ant bin ディレクトリー	<ul style="list-style-type: none"> /usr/lpp/Apache/Ant/apache-ant-1.7.1/bin rsed.envvars - \$ANT_HOME/bin 	
WORKAREA ホーム・ディレクトリー	<ul style="list-style-type: none"> /var/rdz rsed.envvars - \$CGI_ISPWORK 	
SCLMDT プロジェクト構成ホーム・ディレクトリー	<ul style="list-style-type: none"> /var/rdz/sclmdt rsed.envvars - \$_SCLMDT_CONF_HOME 	
ロング/ショート・ネーム変換 VSAM	<ul style="list-style-type: none"> FEK.#CUST.LSTRANS.FILE rsed.envvars - \$_SCLMDT_TRANTABLE 	

WORKAREA および /tmp からの古いファイルの除去

SCLM Developer Toolkit と ISPF の TSO/ISPF クライアント・ゲートウェイは同じ WORKAREA および /tmp ディレクトリーを共用します。これらは両方とも定期的なクリーンアップを必要とする場合があります。このタスクについて詳しくは、154 ページの『(オプション) WORKAREA と /tmp のクリーンアップ』を参照してください。

第 5 章 (オプション) Application Deployment Manager (非推奨)

注: Application Deployment Manager は非推奨になりました。まだサポートされていますが、今後この機能が拡張されることはありません。

Developer for System z は、さまざまなコンポーネントのための共通のデプロイメント方法として、Application Deployment Manager の特定の機能を使用します。以下のいずれかの機能を使用する場合、この章に示すカスタマイズ・ステップが必要です。

- エンタープライズ・サービス・ツール
- BMS Screen Designer
- MFS Screen Designer
- CICSTS コード生成

注: エンタープライズ・サービス・ツールには、サービス・フロー・モデラー (SFM) やエンタープライズ用 XML サービスなどの複数のツールが含まれます。

Application Deployment Manager のカスタマイズでは、CICS リソース定義 (CRD) サーバーが追加されます。このサーバーは、z/OS 上で CICS アプリケーションとして実行されて、以下の機能をサポートします。

- CICS リソース・クエリー
- CICSplex SM 環境および非 CICSplex SM 環境での CICS リソース定義のインストールとアンインストールの要求
- プログラムおよびマップ・セットの段階的導入要求
- パイプライン・スキャン要求
- マニフェストのエクスポート、インポート、および更新要求

CICS 管理者は、「ホスト構成リファレンス」(SA88-4226) の『CICSTS に関する考慮事項』で CRD サーバーの詳細を知ることができます。

要件およびチェックリスト

このカスタマイズ・タスクを完了するには、CICS 管理者、TCP/IP 管理者、およびセキュリティ管理者の支援が必要になります。このタスクには、以下のリソースが特殊なカスタマイズ・タスクが必要です。

- 外部通信用の TCP/IP ポートを定義する
- CICS 領域 JCL を更新する
- CICS 領域 CSD の更新
- CICS 領域に対してグループを定義する
- 管理者に Application Deployment Manager VSAM の更新を許可するセキュリティ規則を作成する
- CICSTS セキュリティーをセットアップする

- (オプション) CICS トランザクション名を定義する
- (オプション) ユーザーに Application Deployment Manager VSAM の更新を許可するセキュリティ規則を作成する

ご使用のサイトで Application Deployment Manager の使用を開始するには、以下のタスクを実行してください。特に断りがない限り、すべてのタスクは必須です。

1. CRD リポジトリを作成します。詳細については、『CRD リポジトリ』を参照してください。
2. 使用する CICS インターフェース (RESTful または Web サービス) を選択します。両インターフェースは共存可能です。詳細については、119 ページの『RESTful と Web サービス』を参照してください。
3. 必要に応じて、REST サービス固有のカスタマイズを行います。詳細については、120 ページの『RESTful インターフェースを使用する CRD サーバー』を参照してください。
 - CICS 主接続領域に対して CRD サーバーを定義します。
 - オプションとして、CICS 非主接続領域に対して CRD サーバーを定義します。
 - オプションとして、CRD サーバー・トランザクション ID をカスタマイズします。
4. 必要に応じて、Web サービス固有のカスタマイズを行います。詳細については、121 ページの『Web サービス・インターフェースを使用する CRD サーバー』を参照してください。
 - パイプライン・メッセージ・ハンドラーを CICS RPL 連結に追加します。
 - CICS 主接続領域に対して CRD サーバーを定義します。
 - オプションとして、CICS 非主接続領域に対して CRD サーバーを定義します。
5. オプションとして、マニフェスト・リポジトリを作成します。詳細については、124 ページの『(オプション) マニフェスト・リポジトリ』を参照してください。

CRD リポジトリ

ADNVCRD ジョブをカスタマイズおよび実行依頼して、CRD リポジトリ VSAM データ・セットの割り振りと初期化を行います。カスタマイズの手順については、メンバー内のドキュメンテーションを参照してください。

ADNVCRD は FEK.#CUST.JCL に置かれます。ただし、FEK.SFEKSAMP(FEKSETUP) ジョブをカスタマイズして実行依頼したときに、別のロケーションを指定した場合は除きます。詳しくは、20 ページの『カスタマイズのセットアップ』を参照してください。

CICS 主接続領域ごとに別々のリポジトリを作成します。リポジトリの共用は、関連するすべての CICS 領域がリポジトリ内に保管された同じ値を使用することを意味します。

注:

- 既存の CRD サーバー・リポジトリを拡張して、Developer for System z バージョン 7.6.1 の管理ユーティリティに追加された URIMAP サポートを使用できるようにする必要があります。詳しくは、「ホスト構成リファレンス」(SA88-4226) の『管理ユーティリティのマイグレーションに関する注』を参照してください。
- 特に断りがない限り、カスタマイズされた値を保持している現行の CRD サーバー・リポジトリは、どの Developer for System z リリースでも再利用できます。

ユーザーには CRD リポジトリへの読み取りアクセス権が必要であり、CICS 管理者には更新アクセス権が必要です。

CICS 管理ユーティリティ

Developer for System z が提供する管理ユーティリティを使用して、CICS 管理者は CICS リソース定義のデフォルト値を指定できます。これらのデフォルトは、読み取り専用とするか、アプリケーション開発者による編集を可能にすることができます。

管理ユーティリティは、ADNJSPAU サンプル・ジョブによって呼び出されます。このユーティリティを使用するには、CRD リポジトリに対する更新アクセス権が必要です。

ADNJSPAU は FEK.#CUST.JCL に置かれます。ただし、FEK.SFEKSAMP(FEKSETUP) ジョブをカスタマイズして実行依頼したときに、別のロケーションを指定した場合は除きます。詳しくは、20 ページの『カスタマイズのセットアップ』を参照してください。

詳細は、「ホスト構成リファレンス」(SA88-4226) の『CICSTS に関する考慮事項』に説明があります。

RESTful と Web サービス

CICS Transaction Server バージョン 4.1 以降は、Representational State Transfer (RESTful) の原則に従って設計された HTTP インターフェースを備えています。現在の RESTful インターフェースは、戦略的な CICSTS インターフェースとしてクライアント・アプリケーションで使用されています。従来の Web サービス・インターフェースはすでに安定化しており、今後は RESTful インターフェースのみが機能拡張の対象となります。

Application Deployment Manager は、この指示書に従い、Developer for System z バージョン 7.6 以降で新たに導入されたすべてのサービスに RESTful CRD サーバーを必要とします。

必要であれば、1 つの CICS 領域で RESTful インターフェースと Web サービス・インターフェースを同時にアクティブにすることができます。この場合、その領域で 2 つの CRD サーバーがアクティブになります。両サーバーは、同じ CRD リポジトリを共有します。2 番目のインターフェースをその領域に対して定義すると、CICS から定義の重複に関する警告が出されます。

RESTful インターフェースを使用する CRD サーバー

このセクションでは、RESTful インターフェースを使用して Developer for System z クライアントと通信する CRD サーバーの定義方法について説明します。

必要であれば、1 つの CICS 領域で RESTful インターフェースと Web サービス・インターフェースを同時にアクティブにすることができます。この場合、その領域で 2 つの CRD サーバーがアクティブになります。両サーバーは、同じ CRD リポジトリを共有します。2 番目のインターフェースをその領域に対して定義すると、CICS から定義の重複に関する警告が出されます。

CICS 主接続領域

CRD サーバーを主接続領域に対して定義する必要があります。主接続領域は、Developer for System z からの Web サービス要求を処理する Web Owning Region (WOR) です。

- FEK.SFEKLOAD(ADNCRD*、ADNANAL、および ADNREST) ロード・モジュールを、CICS 主接続領域の CICS RPL 連結 (DD ステートメント DFHRPL) の中に置いてください。この作業は、適用された保守が自動的に CICS で使用可能になるように、インストール・データ・セットを連結に追加する方法で行ってください。
- ADNCSDRS ジョブをカスタマイズおよび実行依頼して、CICS 主接続領域の CICS システム定義 (CSD) を更新します。カスタマイズの手順については、メンバー内のドキュメンテーションを参照してください。

ADNCSDRS は FEK.#CUST.JCL に置かれます。ただし、FEK.SFEKSAMP(FEKSETUP) ジョブをカスタマイズして実行依頼したときに、別のロケーションを指定した場合は除きます。詳しくは、20 ページの『カスタマイズのセットアップ』を参照してください。

- 適切な CEDA コマンドを使用して、この領域の Application Deployment Manager グループをインストールします。例えば、次のようにします。

```
CEDA INSTALL GROUP(ADNPCRGP)
```

CICS 非主接続領域

CRD サーバーを 1 つ以上の追加の非主接続領域にも使用できます。それらの領域は通常、Application Owning Regions (AOR) です。

注: CICS リソース定義の管理に CICSplex® SM ビジネス・アプリケーション・サービス (BAS) を使用している場合には、これらのステップを実行する必要はありません。

- FEK.SFEKLOAD(ADNCRD*) Application Deployment Manager ロード・モジュールを、これらの非主接続領域の CICS RPL 連結 (DD ステートメント DFHRPL) の中に置きます。この作業は、適用された保守が自動的に CICS で使用可能になるように、インストール・データ・セットを連結に追加する方法で行ってください。
- ADNCSRAR ジョブをカスタマイズおよび実行依頼して、これらの非主接続領域の CSD を更新します。カスタマイズの手順については、メンバー内のドキュメンテーションを参照してください。

ADNCSDAR は FEK.#CUST.JCL にあります。ただし、FEK.SFEKSAMP(FEKSETUP) ジョブをカスタマイズして実行依頼したときに別のロケーションを指定した場合は除きます。詳しくは、20 ページの『カスタマイズのセットアップ』を参照してください。

- 適切な CEDA コマンドを使用して、この領域の Application Deployment Manager グループをインストールします。例えば、次のようにします。

CEDA INSTALL GROUP(ADNARRGP)

(オプション) CRD サーバー・トランザクション ID のカスタマイズ

Developer for System z は、CICS リソースの定義時および照会時に、CRD サーバーが使用する複数のトランザクションを提供します。

表 15. デフォルトの CRD サーバー・トランザクション ID

トランザクション	説明
ADMS	マニフェスト処理ツールからの CICS リソース変更要求用。一般に、これは CICS 管理者が使用するためのものです。
ADMI	CICS リソースを定義、インストール、またはアンインストールする要求用。
ADMR	CICS の環境情報またはリソース情報を取り出す、上記以外のすべての要求用。

トランザクション ID をサイトの標準に合わせて変更することができます。そのためには、以下のステップを実行します。

- ADNTXNC をカスタマイズおよび実行依頼して、ロード・モジュール ADNRCUST を作成します。カスタマイズの手順については、メンバー内のドキュメンテーションを参照してください。
- 結果の ADNRCUST ロード・モジュールを、CRD サーバーが定義されている CICS 領域の CICS RPL 連結 (DD ステートメント DFHRPL) 内に配置します。
- ADNCSDTX をカスタマイズおよび実行依頼して、CRD サーバーが定義されている CICS 領域に対して ADNRCUST をプログラムとして定義します。カスタマイズの手順については、メンバー内のドキュメンテーションを参照してください。

注: RESTful CRD サーバーは、常に ADNRCUST ロード・モジュールをロードしようとし、また、トランザクション ID を変更しない場合でも、ADNRCUST ロード・モジュールを作成して定義しておく、パフォーマンスの面で多少のメリットがあります。

Web サービス・インターフェースを使用する CRD サーバー

このセクションでは、Web サービス・インターフェースを使用して Developer for System z クライアントと通信する CRD サーバーの定義方法について説明します。

必要であれば、1 つの CICS 領域で RESTful インターフェースと Web サービス・インターフェースを同時にアクティブにすることができます。この場合、その領域で 2 つの CRD サーバーがアクティブになります。両サーバーは、同じ CRD リポ

ジトリを共有します。2 番目のインターフェースをその領域に対して定義すると、CICS から定義の重複に関する警告が出されます。

パイプライン・メッセージ・ハンドラー

パイプライン・メッセージ・ハンドラー (ADNTMSGH) は、SOAP ヘッダー内のユーザー ID とパスワードを処理することにより、セキュリティのために使用されます。ADNTMSGH は、サンプルのパイプライン構成ファイルによって参照されるため、CICS RPL 連結の中に入れる必要があります。パイプライン・メッセージ・ハンドラーと必要なセキュリティ・セットアップの詳細については、「[ホスト構成リファレンス](#)」(SA88-4226) の『CICSTS に関する考慮事項』を参照してください。

Developer for System z は、CICS リソースの定義および照会時に、CRD サーバーが使用する複数のトランザクションを提供します。これらのトランザクション ID は、要求された操作に応じて ADNTMSGH が設定します。サイト固有の ADNTMSGH のカスタマイズができるように、サンプルの COBOL ソース・コードが提供されています。

表 16. デフォルトの CRD サーバー・トランザクション ID

トランザクション	説明
ADMS	マニフェスト処理ツールからの CICS リソース変更要求用。一般に、これは CICS 管理者が使用するためのものです。
ADMI	CICS リソースを定義、インストール、またはアンインストールする要求用。
ADMR	CICS の環境情報またはリソース情報を取り出す、上記以外のすべての要求用。

デフォルトの使用:

- FEK.SFEKLOAD(ADNTMSGH) ロード・モジュールを CICS 主接続領域の CICS RPL 連結 (DD ステートメント DFHRPL) の中に置きます。この作業は、適用された保守が自動的に CICS で使用可能になるように、インストール・データ・セットを連結に追加する方法で行ってください。

ADNTMSGH のカスタマイズ:

サンプルのメンバー ADNMSGH* は、FEK.#CUST.JCL および FEK.#CUST.COBOL に置かれます。ただし、FEK.SFEKSAMP(FEKSETUP) ジョブをカスタマイズして実行依頼したときに別のロケーションを指定した場合は除きます。詳しくは、20 ページの『カスタマイズのセットアップ』を参照してください。

- サンプルのパイプライン・メッセージ・ハンドラー (COBOL) ソース・コード FEK.#CUST.COBOL(ADNMSGHS) を、ご使用のサイトの標準に合わせてカスタマイズします。
- FEK.#CUST.JCL(ADNMSGHC) ジョブをカスタマイズおよび実行依頼して、カスタマイズされた ADNMSGHS ソースをコンパイルします。カスタマイズの手順については、ADNMSGHC 内のドキュメンテーションを参照してください。結果のロード・モジュールの名前は、ADNTMSGH とする必要があります。

- 結果の ADNTMSGH ロード・モジュールを CICS 主接続領域の CICS RPL 連結 (DD ステートメント DFHRPL) の中に置きます。

注: 必ず、カスタマイズした ADNTMSGH ロード・モジュールを FEK.SFEKLOAD への参照の前に配置してください。そうしなかった場合は、デフォルトのロード・モジュールが使用されます。

CICS 主接続領域

CRD サーバーを主接続領域に対して定義する必要があります。主接続領域は、Developer for System z からのサービス要求を処理する領域です。

- FEK.SFEKLOAD(ADNCRD*, ADNANAL、および ADNREST) ロード・モジュールを、CICS 主接続領域の CICS RPL 連結 (DD ステートメント DFHRPL) の中に置いてください。この作業は、適用された保守が自動的に CICS で使用可能になるように、インストール・データ・セットを連結に追加する方法で行ってください。122 ページの『パイプライン・メッセージ・ハンドラー』で説明しているように、パイプライン・メッセージ・ハンドラー・ロード・モジュール ADNTMSGH も RPL 連結の中に置く必要があります。
- ADNCSDWS ジョブをカスタマイズおよび実行依頼して、CICS 主接続領域の CICS システム定義 (CSD) を更新します。カスタマイズの手順については、メンバー内のドキュメンテーションを参照してください。このジョブで使用するトランザクション ID は、パイプライン・メッセージ・ハンドラー (カスタマイズされている場合があります) で使用するトランザクション ID と一致しなければなりません。

ADNCSDWS は FEK.#CUST.JCL に置かれます。ただし、FEK.SFEKSAMP(FEKSETUP) ジョブをカスタマイズして実行依頼したときに、別のロケーションを指定した場合は除きます。詳しくは、20 ページの『カスタマイズのセットアップ』を参照してください。

- 適切な CEDA コマンドを使用して、この領域の Application Deployment Manager グループをインストールします。例えば、次のようにします。

```
CEDA INSTALL GROUP(ADNPCRGP)
```

CICS 非主接続領域

CRD サーバーを 1 つ以上の追加の非主接続領域にも使用できます。それらの領域は通常、Application Owning Regions (AOR) です。

注: CICS リソース定義の管理に CICSplex SM ビジネス・アプリケーション・サービス (BAS) を使用している場合には、これらのステップを実行する必要はありません。

- FEK.SFEKLOAD(ADNCRD*) Application Deployment Manager ロード・モジュールを、これらの非主接続領域の CICS RPL 連結 (DD ステートメント DFHRPL) の中に置きます。この作業は、適用された保守が自動的に CICS で使用可能になるように、インストール・データ・セットを連結に追加する方法で行ってください。
- ADNCSDAR ジョブをカスタマイズおよび実行依頼して、これらの非主接続領域の CSD を更新します。カスタマイズの手順については、メンバー内のドキュメンテーションを参照してください。

ADNCSDAR は FEK.#CUST.JCL にあります。ただし、FEK.SFEKSAMP(FEKSETUP) ジョブをカスタマイズして実行依頼したときに別のロケーションを指定した場合は除きます。詳しくは、20 ページの『カスタマイズのセットアップ』を参照してください。

- 適切な CEDA コマンドを使用して、この領域の Application Deployment Manager グループをインストールします。例えば、次のようにします。

CEDA INSTALL GROUP(ADNARRGP)

(オプション) マニフェスト・リポジトリ

Developer for System z では、クライアントは選択した CICS リソースを記述しているマニフェストを参照でき、オプションとして、変更することもできます。CICS 管理者によって設定された許可に応じて、変更を直接行うか、あとで CICS 管理者が処理できるよう、マニフェスト・リポジトリにエクスポートすることができます。

注:

- このステップが必要になるのは、マニフェスト処理ツールで処理するためにマニフェストを Developer for System z からエクスポートする場合に限られます。
- マニフェスト処理ツールは、IBM CICS エクスプローラー用のプラグインです。

マニフェスト・リポジトリ VSAM データ・セットの割り振りと初期化を行い、それを CICS 主接続領域に対して定義するために、ADNVMFST ジョブをカスタマイズして実行依頼します。カスタマイズの手順については、メンバー内のドキュメンテーションを参照してください。CICS 主接続領域ごとに別々のマニフェスト・リポジトリを作成する必要があります。すべてのユーザーにマニフェスト・リポジトリに対する更新アクセス権が必要です。

ADNVMFST は FEK.#CUST.JCL に置かれます。ただし、FEK.SFEKSAMP(FEKSETUP) ジョブをカスタマイズして実行依頼したときに、別のロケーションを指定した場合は除きます。詳しくは、20 ページの『カスタマイズのセットアップ』を参照してください。

第 6 章 (オプション) ホスト・ベースのコード分析

Developer for System z クライアントと同様に、Developer for System z ホストは別の製品として提供されている実行中のコード分析ツール、IBM Rational Developer for System z Host Utilities をサポートします。このコード分析をホスト上で実行する利点は、それを日常のバッチ処理として組み込むことができる点です。

以下のコード分析ツールを、ホスト上で使用することができます。

- コード・レビュー: コード・レビューは、異なる重大度レベルを持つ規則を使用して、ソース・コードをスキャンし、規則違反を報告します。
- コード・カバレッジ: 実行中のプログラムを分析し、実行可能な行の総数に対する実行済みの行についてのレポートを生成します。

要件およびチェックリスト

ホスト・ベースのコード分析ツールをご使用のサイトで使用開始する際には、他の管理者の支援は必要ありません。ただし、以下のタスクを実行する必要があります。特に断りがない限り、すべてのタスクは必須です。

1. IBM Rational Developer for System z Host Utilities をインストールします
(「*Program Directory for IBM Rational Developer for System z Host Utilities*」(GI88-4326) を参照)。提供されているデフォルトを使用すると、製品は高位修飾子の AKG および z/OS UNIX パス/usr/lpp/rdzutil を使用してインストールされます。
2. AKG.SAKGSAMP(AKGSETUP) をカスタマイズして実行依頼することで、提供されているサンプルのカスタマイズ可能なコピーを作成します。このジョブは、以下のタスクを実行します。
 - AKG.#CUST.PROCLIB を作成し、これにサンプル SYS1.PROCLIB メンバーを取り込みます。
 - AKG.#CUST.JCL を作成し、これにサンプル構成 JCL を取り込みます。

注: IBM Debug Tool for z/OS は、コード・カバレッジ・コンポーネントの前提条件です。

コード・レビュー

コード・レビューは、ソース・コードをスキャンし、異なる重大度レベルを持つ規則を使用して、規則違反を報告します。ツールには COBOL および PL/I の規則プロバイダーが付属していますが、それ以外の規則プロバイダーを追加することもできます。

Developer for System z Host Utilities は、バッチ・モードでのコード・レビュー・サービスの呼び出しを簡素化するために、サンプル・プロシージャ AKGCR を提供しています。AKGCR は AKG.#CUST.PROCLIB に置かれます。ただし、AKG.SAKGSAMP(AKGSETUP) ジョブをカスタマイズして実行依頼したときに別のロケーションを指定した場合は除きます。

サンプル・プロシージャ AKG.#CUST.PROCLIB(AKGCR) を、メンバー内で説明されているようにカスタマイズし、SYS1.PROCLIB にコピーします。

AKGCR プロシージャをシステム・プロシージャ・ライブラリーにコピーできない場合は、呼び出しジョブに JCLLIB カードを JOB カードの直後に追加するように Developer for System z ユーザーに依頼します。

```
//MYJOB      JOB <job parameters>
//PROCS      JCLLIB ORDER=(AKG.#CUST.PROCLIB)
```

コード・レビュー・プロセスの変更

Developer for System z コード・レビューは、サード・パーティーのコードを、レビュー・プロセスの一部にすることを可能にします。例えば、C/C++ コードを分析する規則プロバイダーを提供したり、サイト固有のコーディング規則を認識するよう COBOL 規則プロバイダーを拡張したりすることができます。

ホスト・ベースのコード・レビューは、Developer for System z クライアントと同じく Eclipse プロセスです。そのため、開発サポート・チームがクライアント上でコード・レビュー用に行った機能拡張は、ホスト上でも再利用することができます。

機能拡張は、Eclipse プラグインまたは Eclipse フィーチャーで構成されます。それらをアクティブ化するには、AKGCRADD 構成ジョブで記述されているように、それらを既存のコードで使用できるようにします。AKGCRADD は AKG.#CUST.JCL に置かれます。ただし、AKG.SAKGSAMP(AKGSETUP) ジョブをカスタマイズして実行依頼したときに別のロケーションを指定した場合は除きます。

コード・カバレッジ

| コード・カバレッジは、実行中のプログラムを分析し、実行可能な行の総数に対する
| 実行済みの行についてのレポートを生成します。コード・カバレッジは、一時ポ
| ートを使用して、IBM Debug Tool for z/OS との TCP/IP 接続をセットアップする
| ことにご注意ください。

| Developer for System z Host Utilities には、コード・カバレッジをバッチ・モード
| で呼び出す 2 とおりの方法があります。単一のプログラム実行を処理するサンプル
| JCL プロシージャを使用する方法と、複数のプログラム実行を処理できる永続的
| にアクティブなコード・カバレッジ・コレクターを開始および停止するためのスク
| リプト・セットを使用する方法です。

コード・カバレッジの単一の呼び出し

AKGCC サンプル・プロシージャは、コード・カバレッジ・コレクターを開始して、単一プログラム実行を分析させ、コレクターを停止し、結果をアーカイブして後で利用できるようにするための手段となります。

AKGCC は AKG.#CUST.PROCLIB に置かれます。ただし、AKG.SAKGSAMP(AKGSETUP) ジョブをカスタマイズして実行依頼したときに別のロケーションを指定した場合は除きます。

サンプル・プロシージャ AKG.#CUST.PROCLIB(AKGCC) を、メンバー内で説明されているようにカスタマイズし、SYS1.PROCLIB にコピーします。

AKGCC プロシージャーをシステム・プロシージャー・ライブラリーにコピーできない場合、JCLLIB カード (JOB カードの直後) を呼び出しジョブに追加するように Developer for System z ユーザーに依頼します。

```
//MYJOB      JOB <job parameters>
//PROCS      JCLLIB ORDER=(AKG.#CUST.PROCLIB)
```

コード・カバレッジの複数の呼び出し

AKGCC プロシージャーにはすべてをこれで処理できる利便性がありますが、ソフトウェア開発プロセスのセットアップ方法によっては、プログラム分析のたびにリソースと時間を使ってコード・カバレッジ・コレクターを開始しなければならないことの方が重大かもしれません。

Developer for System z Host Utilities には、アクティブな状態を維持するコード・カバレッジ・コレクターを開始するための `ccstart` スクリプトが用意されています。開始したコレクターを、コード・カバレッジの複数の呼び出しで使用できます。コレクターを停止するには、`ccstop` スクリプトを使用できます。

これらのスクリプト (`ccstart`、`ccstop`、および `codecov`) は、Developer for System z Host Utilities をデフォルトのロケーションにインストールした場合、`/usr/lpp/rdzutil/bin/` にあります。これらのスクリプトの使用法については、Developer for System z IBM Knowledge Center を参照してください。

一般的な使用シナリオを以下に示します。

1. 固定ポート番号を使用するオプションと出力を既知のロケーションにリダイレクトするオプションを指定して `ccstart` を呼び出します。
2. 始動オプションで、分析対象のプログラムを必要な回数呼び出します。
`TEST(,,,TCPIP&<hostip>:<port>)`。
3. `ccstop` を呼び出します。
4. Developer for System z クライアントから出力ロケーションにアクセスして、レポートを表示します。

コード・カバレッジの出力

コード・カバレッジの出力は、Developer for System z クライアントにインポートすることを目的としているため、z/OS UNIX ファイルに書き込まれます。またコード・カバレッジは、以前の実行結果を使用し、それらを現在の実行結果と組み合わせて複数のコード・パスをカバーする単一のレポートを作成することができます。

これらの理由により、Developer for System z Host Utilities はコード・カバレッジの実行の出力を削除しようとしなため、出力は長期にわたって累積されます。

z/OS UNIX には、ファイルが入っているディレクトリーとファイルの経過日数に基づいてファイルを削除する、`skulker` というシェル・スクリプトがあります。指定された日時にコマンドを実行する z/OS UNIX cron デーモンと結合すれば、定期的にターゲット・ディレクトリーを空にする自動化ツールをセットアップできます。`skulker` スクリプトおよび cron デーモンの詳細については、「UNIX System Services コマンド解説書」(SA88-8641) を参照してください。

第 7 章 (オプション) その他のカスタマイズ・タスク

このセクションは、さまざまなオプションのカスタマイズ・タスクを結合したものです。必要なサービスを構成するには、該当するセクションの説明に従ってください。

Developer for System z 構成ファイルへのカスタマイズ

- 『(オプション) pushtoclient.properties、ホスト・ベースのクライアント制御』
- 133 ページの 『(オプション) ssl.properties、RSE SSL 暗号化』
- 135 ページの 『(オプション) rsecomm.properties、RSE トレース』
- 137 ページの 『(オプション) include.conf、C/C++ コンテンツ・アシスト用の強制インクルード』

他の製品への Developer for System z 関連のカスタマイズ :

- 138 ページの 『(オプション) z/OS UNIX サブプロジェクト』
- 139 ページの 『(オプション) インクルード・プリプロセッサのサポート』
- 140 ページの 『(オプション) Enterprise COBOL および PL/I での xUnit サポート』
- 141 ページの 『(オプション) エンタープライズ・サービス・ツール・サポート』
- 142 ページの 『(オプション) CICS 双方向言語サポート』
- 143 ページの 『(オプション) 生成コードの診断用 IRZ メッセージ』
- 144 ページの 『(オプション) 統合デバッガー』
- 152 ページの 『(オプション) 問題判別ツールのサポート』
- 152 ページの 『(オプション) DB2 および IMS のデバッグのサポート』
- 153 ページの 『(オプション) File Manager のサポート』
- 154 ページの 『(オプション) WORKAREA と /tmp のクリーンアップ』

(オプション) pushtoclient.properties、ホスト・ベースのクライアント制御

このカスタマイズ・タスクには、支援や特殊リソース、または基本セットアップのための特殊なカスタマイズ・タスクは必要ありません。

グループ・サポートを使用可能にする場合、このカスタマイズ・タスクを完了するには、セキュリティ管理者または LDAP 管理者の支援が必要になります。このタスクには、以下のリソースまたは特殊なカスタマイズ・タスクが必要です。

- FEK.PTC.* プロファイルへのユーザー・アクセスを許可するセキュリティ規則
- または、FEK.PTC.* LDAP グループのユーザー・メンバーシップの定義

Developer for System z クライアント・バージョン 8.0.1 以降では、接続時にホスト・システムからクライアント構成ファイルと製品の更新情報を取り出すことができます。そのため、すべてのクライアントで最新の設定を共有できます。

z/OS プロジェクトは、クライアント上の「z/OS プロジェクト」パースペクティブを通じて個別に定義するか、ホスト・システム上で集中的に定義してクライアントへユーザー単位で伝搬することができます。それらのホスト・ベースのプロジェクトは、クライアント上で定義されたプロジェクトと外観も機能もまったく同じですが、クライアントからは構造、メンバー、およびプロパティーを変更できず、また、ホスト・システムに接続しないとアクセスできない点が異なっています。

`pushtoclient.properties` はクライアントに対し、これらの機能が使用可能であるか、関連のデータがどこに保管されているかを伝達します。このデータは、Developer for System z クライアント管理者または開発プロジェクト・マネージャーが保守します。

`pushtoclient.properties` は `/etc/rdz/` に置かれます。ただし、FEK.SFEKSAMP(FEKSETUP) ジョブをカスタマイズして実行依頼したときに、別のロケーションを指定した場合は除きます。詳しくは、20 ページの『カスタマイズのセットアップ』を参照してください。このファイルは、TSO **OEDIT** コマンドで編集できます。変更を有効にするには、RSED 開始タスクを再始動してください。

バージョン 8.0.3 以降、クライアント管理者は、さまざまな開発者グループのニーズに合わせて、複数のクライアント構成セットおよび複数のクライアント更新シナリオを作成することができます。これらの複数のセットおよびシナリオを使用して、LDAP グループのメンバーシップやセキュリティ・プロファイルに対する許可などの基準に基づいて、カスタマイズされたセットアップをユーザーに提供できます。複数のグループをサポートする方法の詳細については、「ホスト構成リファレンス」(SA88-4226) の『クライアントへのプッシュの考慮事項』を参照してください。

次のコード・サンプルは `pushtoclient.properties` ファイルを示しています。このファイルは、使用するシステム環境に合わせてカスタマイズする必要があります。US コード・ページを使用する場合、コメント行は番号記号 (#) で始まります。データ行には、ディレクティブとそれに割り当てられる値のみを入れることができます。同じ行にコメントを入れることはできません。行の継続はサポートされていません。

```
#
# host-based client control
#
config.enabled=false
product.enabled=false
reject.config.updates=false
reject.product.updates=false
accept.product.license=false
primary.system=false
pushtoclient.folder=/var/rdz/pushtoclient
default.store=com.ibm.ftt.configurations.USS
file.permission=RWX.RWX.RX
```

図 35. `pushtoclient.properties`: ホスト・ベースのクライアント制御構成ファイル

config.enabled

構成ファイルに対してホスト・ベースのクライアント制御を使用するかどうか

かを示します。デフォルトは `false` です。有効な値は `true`、`false`、LDAP、または SAF です。これらの値の意味については、132 ページの表 17 を参照してください。

product.enabled

製品更新に対してホスト・ベースのクライアント制御を使用するかどうかを示します。デフォルトは `false` です。有効な値は `true`、`false`、LDAP、または SAF です。これらの値の意味については、132 ページの表 17 を参照してください。

reject.config.updates

クライアントにプッシュされた構成更新をユーザーが拒否できるかどうかを示します。デフォルトは `false` です。有効な値は `true`、`false`、LDAP、または SAF です。これらの値の意味については、132 ページの表 17 を参照してください。

reject.product.updates

クライアントにプッシュされた製品更新をユーザーが拒否できるかどうかを示します。デフォルトは `false` です。有効な値は `true`、`false`、LDAP、または SAF です。これらの値の意味については、132 ページの表 17 を参照してください。

accept.product.license

クライアントへのプッシュによって開始した更新中に、製品ライセンスが自動的に受諾されるかどうかを示します。有効になっている場合、IBM Installation Manager で、クライアント更新中にライセンスを受け入れるように求められることはありません。デフォルトは `false` です。有効な値は、`true` と `false` だけです。

primary.system

ホスト・ベースのクライアント制御は、システム固有のデータをシステムごとに保管しながら、共通データを単一のシステム上で保守できるようにして、管理上の負担を軽減します。このディレクティブは、これがグローバルな、システム固有でないクライアント定義を保管するシステムであるかどうかを示します。デフォルトは `false` です。有効な値は、`true` と `false` だけです。

注: 1 つのシステムのみを 1 次システムとして定義するようにしてください。Developer for System z クライアント管理者は、ターゲット・システムが 1 次システムでない場合、グローバル構成データをエクスポートできません。Developer for System z クライアントは、構成が同期していない複数の 1 次システムに接続する場合、不規則な動作を示すことがあります。

pushtoclient.folder

ホスト・ベースのクライアント制御定義の基本ディレクトリー。デフォルトは `/var/rdz/pushtoclient` です。

default.store

ホスト・ベースのクライアント制御は、クライアントにプッシュされるデータを保管するためのさまざまな方式をサポートしています。このディレクティブは、データにアクセスするために使用するドライバー、すなわちストア

を識別します。デフォルトは `com.ibm.ftt.configurations.USS` で、これは z/OS UNIX フラット・ファイルに保管されているデータをサポートします。

Developer for System z が提供するのは、`com.ibm.ftt.configurations.USS` ストアのみです。データを他の場所に配置する場合、サード・パーティーのストアが必要です。

file.permission

`com.ibm.ftt.configurations.USS` ストアでは、`file.permission` を使用して、このストアで作成されたファイルに必要なアクセス権を決定します。デフォルトは `RWX.RWX.RX` で、所有者および所有者のデフォルト・グループに、ディレクトリー構造および構造内のファイルの読み取り/書き込みアクセス権を許可します。他のすべてのユーザーは、ディレクトリー構造および構造内のファイルの読み取りアクセス権のみを持ちます。

UNIX 標準に従った場合、所有者、グループ、およびその他という 3 つのタイプのユーザーに対してアクセス権を設定できます。`file.permission` マスクのフィールドは、この順序に一致しており、ピリオド (.) で区切られます。各フィールドは、空にするか、`R`、`W`、`RW`、`X`、`RX`、`WX`、または `RWX` を値として設定できます (ここで、`R` = 読み取り、`W` = 書き込み、`X` = 実行またはディレクトリーの内容をリストします)。

表 17. クライアントへのプッシュのグループ・サポート

キー値	関連するクライアントへのプッシュ機能が有効になっているか。
False	いいえ、無効です
True	はい、すべて有効です
LDAP	はい。ただし、可用性は LDAP グループのメンバーシップによって制御されます
SAF	はい。ただし、可用性はセキュリティー・プロファイルに対する許可によって制御されます

注:

- ホスト・ベースのクライアント制御をアクティブにするには、`keymapping.xml` ファイルが `/var/rdz/pushtoclient` に存在する必要があります。このファイルは、Developer for System z クライアント管理者によって作成され保守されます。
- ホスト・ベースのプロジェクト、ホスト・ベースのクライアント構成、およびアップグレード制御の詳細については、「ホスト構成リファレンス」(SA88-4226) の『クライアントへのプッシュの考慮事項』を参照してください。
- ファイルの作成時に、z/OS UNIX は、デフォルトで作成スレッドの実効 UID (ユーザー ID) と所有ディレクトリーの GID (グループ ID) を使用します。作成スレッドの実効 GID は使用しません。この動作を変更する方法、または必要な GID 割り当てが行われるようにホスト・ベースのクライアント制御セットアップを調整する方法の詳細については、「ホスト構成リファレンス」(SA88-4226) の『z/OS UNIX ディレクトリー構造』を参照してください。

(オプション) ssl.properties、RSE SSL 暗号化

このカスタマイズ・タスクを完了するには、セキュリティー管理者の支援が必要になります。このタスクには、以下のリソースまたは特殊なカスタマイズ・タスクが必要です。

- LINKLIST 更新
 - プログラム制御データ・セットを追加するセキュリティー規則
 - (オプション) SSL の証明書を追加するセキュリティー規則
-

外部のクライアント/ホスト通信を SSL (Secure Socket Layer) で暗号化できます。このフィーチャーは、デフォルトでは使用不可に設定され、`ssl.properties` 内の設定によって制御されます。

注: X.509 証明書によるクライアント認証では、SSL 暗号化通信を使用する必要があります。

`ssl.properties` は `/etc/rdz/` に置かれます。ただし、`FEK.SFEKSAMP(FEKSETUP)` ジョブをカスタマイズして実行依頼したときに、別のロケーションを指定した場合は除きます。詳しくは、20 ページの『カスタマイズのセットアップ』を参照してください。このファイルは、`TSO OEDIT` コマンドで編集できます。変更を有効にするには、RSE の再始動が必要です。

クライアントは接続のセットアップ時には RSE デーモンと通信し、実際のセッションのときは RSE サーバーと通信します。どちらのデータ・ストリームも、SSL を使用可能にした場合は暗号化されます。

RSE デーモンおよび RSE サーバーは、両者間のアーキテクチャーの違いから、証明書の保管に関して異なるメカニズムをサポートしています。これは、RSE デーモンと RSE サーバーの両方に SSL 定義が必要であることを意味しています。RSE デーモンと RSE サーバーが同じ証明書管理方式を使用する場合は、共用証明書を使用できます。

表 18. SSL 証明書の保管メカニズム

証明書ストレージ	作成者および管理者	RSE デーモン	RSE サーバー
鍵リング	SAF 準拠のセキュリティー製品	サポート対象	サポート対象
鍵データベース	z/OS UNIX の gskkyman	サポート対象	/
鍵ストア	Java の keytool	/	サポート対象

注:

- SAF 準拠の鍵リングは、証明書の管理に推奨される方式です。
- SAF 準拠の鍵リングでは、証明書の秘密鍵が、セキュリティー・データベースに保管されるか、または System z 暗号化ハードウェアとのインターフェースである ICSF を使用して保管されます。ICSF へのアクセスは、CSFSERV セキュリティー・クラス内のプロファイルによって保護されます。

RSE デーモンは、System SSL の機能を使用して SSL を管理します。これは SYS1.SIEALNKE が、セキュリティー・ソフトウェアによってプログラム制御されることが必要で、LINKLIST または rsed.envvars 内の STEPLIB ディレクティブを使用する場合は RSE から使用可能でなければならないことを意味しています。

次のコード・サンプルは ssl.properties ファイルを示しています。このファイルは、使用するシステム環境に合わせてカスタマイズする必要があります。US コード・ページを使用する場合、コメント行は番号記号 (#) で始まります。データ行には、ディレクティブとそれに割り当てられる値のみを入れることができ、その同じ行にコメントを入れることはできません。行の継続はサポートされていません。

```
# ssl.properties - SSL configuration file
enable_ssl=false

# Daemon Properties

#daemon_keydb_file=
#daemon_keydb_password=
#daemon_key_label=

# Server Properties

#server_keystore_file=
#server_keystore_password=
#server_keystore_label=
#server_keystore_type=JCECERACFKS
```

図 36. ssl.properties - SSL 構成ファイル

デーモンおよびサーバー・プロパティを設定する必要があるのは、SSL を使用可能にする場合のみです。SSL のセットアップについて詳しくは、「*Developer for System z* ホスト構成リファレンス」の『SSL および X.509 認証のセットアップ』を参照してください。

enable_ssl

SSL 通信を使用可能または使用不可に設定します。デフォルトは false です。有効な設定値は、true と false だけです。

daemon_keydb_file

RACF または同様のセキュリティー製品の鍵リング名。鍵リングを使用せず、gskkyman を使用して鍵データベースを作成した場合は、鍵データベース名を指定してください。SSL を使用可能にする場合は、このディレクティブのコメントを外してカスタマイズします。

daemon_keydb_password

鍵リングを使用する場合は、コメント化されたままにするか、ブランクにします。そうでない場合は、鍵データベースのパスワードを指定します。SSL を使用可能にし、gskkyman 鍵データベースを使用する場合は、このディレクティブのコメントを外してカスタマイズします。

daemon_key_label

鍵リングまたは鍵データベースで使用される証明書ラベル (デフォルトのものとして定義されていない場合)。デフォルトを使用する場合は、コメント化する必要があります。SSL を使用可能にするが、デフォルトのセキュリティー証明書は使用しない場合、このディレクティブのコメントを外してカスタマイズします。鍵ラベルでは、大/小文字が区別されます。

server_keystore_file

Java の **keytool** コマンドによって作成された鍵ストアの名前、または、**server_keystore_type=JCERACFKS** の場合は **RACF** または同様のセキュリティー製品の鍵リング名。SSL を使用可能にする場合は、このディレクティブのコメントを外してカスタマイズします。

server_keystore_password

鍵リングを使用する場合は、コメント化されたままにするか、ブランクにします。そうでない場合は、鍵ストアのパスワードを指定します。SSL を使用可能にし、**keytool** 鍵ストアを使用する場合は、このディレクティブのコメントを外してカスタマイズします。

server_keystore_label

鍵リングまたは鍵ストアで使用される証明書ラベル。デフォルトは、最初に検出された有効な証明書です。SSL を使用可能にするが、デフォルトのセキュリティー証明書は使用しない場合、このディレクティブのコメントを外してカスタマイズします。鍵ラベルでは、大/小文字が区別されます。

server_keystore_type

鍵ストアのタイプ。デフォルトは **JKS** です。有効な値は、以下のとおりです。

表 19. 有効な鍵ストアのタイプ

キーワード	鍵ストアのタイプ
JKS	Java 鍵ストア
JCERACFKS	SAF 準拠の鍵リング。この場合は、証明書の秘密鍵がセキュリティー・データベースに保管されます。
JCECCARACFKS	SAF 準拠の鍵リング。この場合は、証明書の秘密鍵が System z 暗号化ハードウェアとのインターフェースである ICSF を使用して保管されます。

注: 本書を公開した時点では、IBM z/OS Java で JCECCARACFKS をサポートするためには、`/usr/lpp/java/J6.0/lib/security/java.security` ファイルを更新する必要があります。以下の行を追加する必要があります。

```
security.provider.1=com.ibm.crypto.hdwrCCA.provider.IBMJCECCA
```

更新後のファイルは以下のようになります。

```
security.provider.1=com.ibm.crypto.hdwrCCA.provider.IBMJCECCA
security.provider.2=com.ibm.jsse2.IBMJSSEProvider2
security.provider.3=com.ibm.crypto.provider.IBMJCE
security.provider.4=com.ibm.security.jgss.IBMJGSSProvider
security.provider.5=com.ibm.security.cert.IBMCertPath
security.provider.6=com.ibm.security.sasl.IBMSASL
```

(オプション) rsecomm.properties、RSE トレース

このカスタマイズ・タスクには、支援や特殊リソース、または特殊なカスタマイズ・タスクは必要ありません。

Developer for System z は、問題解決の目的から、さまざまなレベルでの内部プログラム・フローのトレースをサポートしています。RSE、および RSE が呼び出すサービスの一部では、出力ログでの必要な初期の詳細レベルを認識するために、`rsecomm.properties` 内の設定を使用します。

重要: これらの設定の変更は、パフォーマンス低下の原因になるため、IBM サポートの指示の下でのみ実行してください。

`rsecomm.properties` は `/etc/rdz/` に置かれます。ただし、FEK.SFEKSAMP (FEKSETUP) ジョブをカスタマイズして実行依頼したときに、別のロケーションを指定した場合は除きます。詳しくは、20 ページの『カスタマイズのセットアップ』を参照してください。このファイルは、TSO **OEDIT** コマンドで編集できます。

次のコード・サンプルは、`rsecomm.properties` ファイルを示しています。このファイルは、トレースの必要性に合わせてカスタマイズできます。US コード・ページを使用する場合、コメント行は番号記号 (#) で始まります。データ行には、ディレクティブとそれに割り当てられる値のみを入れることができ、その同じ行にコメントを入れることはできません。行の継続はサポートされていません。

```
# server.version - DO NOT MODIFY!
server.version=5.0.0

# Logging level
# 0 - Log error messages
# 1 - Log error and warning messages
# 2 - Log error, warning and info messages
debug_level=1
#USER=userid
#USER=(userid,userid,...)
```

図 37. `rsecomm.properties` - ロギング構成ファイル

server.version
ロギング・サーバー・バージョン。デフォルトは 5.0.0 です。変更しないでください。

debug_level
出力ログの詳細レベル。デフォルトは 1 です。この場合、エラー・メッセージおよび警告メッセージがログに記録されます。`debug_level` により複数のサービスの詳細レベルが (したがって複数の出力ファイルが) 制御されます。詳細レベルの増大は、パフォーマンス低下の原因になるため、IBM サポートの指示の下でのみ実行してください。このディレクティブによってどのログが制御されるかについて詳しくは、「[ホスト構成リファレンス](#)」(SA88-4226) の『RSE トレース』を参照してください。
有効な値は以下のとおりです。

0	エラー・メッセージのみをログに記録します。
1	エラー・メッセージと警告メッセージをログに記録します。
2	エラー・メッセージ、警告メッセージ、および情報メッセージをログに記録します。

注: `debug_level` は、221 ページの『第 11 章 オペレーター・コマンド』に記載されているとおり、`modify rsecommlog`、`modify rseserverlog`、および `modify rsedaemonlog` のオペレーター・コマンドを使用して特定のログ・ファイルについて動的に変更できます。

USER サーバー・セットアップ中に、指定されたユーザー ID に関してデバッグ・レベル 2 (ログ・エラー、警告、および通知メッセージ) を設定します。他のすべてのユーザーに関するデバッグ・レベルは、`debug_level` ディレクティブで指定されたデフォルトです。USER ディレクティブは RSE サーバー (`rsecomm.log`) および MVS データ・セット・サービス (`lock.log` と `ffs*.log`) のトレース詳細レベルを変更します。これは `modify trace user` オペレーター・コマンドを発行することと同等です。

(オプション) `include.conf`、C/C++ コンテンツ・アシスト用の強制インクルード

このカスタマイズ・タスクには、支援や特殊リソース、または特殊なカスタマイズ・タスクは必要ありません。

C/C++ のコンテンツ・アシストは、`include.conf` の定義を使用して、指定されたファイルまたはメンバーの強制インクルードを実行できます。強制インクルードは、ファイルまたはメンバーがプリプロセッサ・ディレクティブを使用してソース・コードに組み込まれたかどうかに関係なく、コンテンツ・アシスト操作の実行時に構文解析されるファイルまたはディレクトリー、データ・セット、またはデータ・セット・メンバーで構成されます。

ファイルは、使用される前に、`rsed.envvars` で `include.c` 変数または `include.cpp` 変数によって参照される必要があります。このように `rsed.envvars` で参照されることは、C および C++ で使用されるように別のファイルを指定できることを意味します。`rsed.envvars` の変数はデフォルトで使用不可になっています。

サンプルの `include.conf` は `/etc/rdz/` に置かれます。ただし、ジョブ `FEK.SFEKSAMP(FEKSETUP)` をカスタマイズして実行依頼したときに、別のロケーションを指定した場合は除きます。詳細については、20 ページの『カスタマイズのセットアップ』を参照してください。このファイルは、TSO **OEDIT** コマンドで編集できます。

定義は列 1 から始める必要があります。US コード・ページを使用する場合、コメント行はポンド記号 (#) で始まります。データ行に入れることができるのは、ディレクトリー、ファイル、データ・セット、またはメンバーの名前だけです。同じ行にコメントを入れることはできません。行の継続はサポートされていません。


```

# To include the stdio.h file from the /usr/include directory, input:
# /usr/include/stdio.h
#
# To include all files of the /usr/include directory and all of it's
# sub-directories, input:
# /usr/include
#
# Uncomment and customize variable FILETYPES to limit the z/OS UNIX
# wildcard include to selected (case sensitive) file types:
# The file types are specified in a comma-delimited list (no blanks)
# FILETYPES=H,h,hpp,C,c,cpp,cxx

# To include all members of the CBC.SCLBH.H data set, input:
# //CBC.SCLBH.H
#
# To include the STDIOSTR member of the CBC.SCLBH.H data set, input:
# //CBC.SCLBH.H(STDIOSTR)
# The sample list contains some commonly used C standard library files
/usr/include/assert.h
/usr/include/ctype.h
/usr/include/errno.h
/usr/include/float.h
/usr/include/limits.h
/usr/include/locale.h
/usr/include/math.h
/usr/include/setjmp.h
/usr/include/signal.h
/usr/include/stdarg.h
/usr/include/stddef.h
/usr/include/stdio.h
/usr/include/stdlib.h
/usr/include/string.h
/usr/include/time.h

```

図 38. *include.conf* - C/C++ コンテンツ・アシスト用の強制インクルード

(オプション) z/OS UNIX サブプロジェクト

このカスタマイズ・タスクには、支援や特殊リソース、または特殊なカスタマイズ・タスクは必要ありません。

REXEC (リモート実行) は、クライアントがホスト・システム上でコマンドを実行できるようにする TCP/IP サービスです。SSH (セキュア・シェル) も同様のサービスですが、すべての通信が SSL (Secure Socket Layer) によって暗号化されます。Developer for System z は、いずれかのサービスを使用して、z/OS UNIX サブプロジェクトでリモート (ホスト・ベース) アクションを実行します。

注:

- Developer for System z は、TSO パージョンでなく、z/OS UNIX パージョンの REXEC を使用します。
- REXEC/SSH がデフォルト・ポートを使用するように構成されていない場合、Developer for System z クライアントは z/OS UNIX サブプロジェクトに使用する正しいポートを定義する必要があります。この構成を行うには、「ウィンドウ」 > 「設定」 > 「z/OS ソリューション」 > 「USS サブプロジェクト」 > 「リモ

ート・アクション・オプション」設定ページを選択します。使用するポートについては、『REXEC または SSH のセットアップ』を参照してください。

REXEC または SSH のセットアップ

REXEC および SSH は、別の TCP/IP サービスである、INETD (Internet Daemon) によって提供されるサービスに依存します。INETD、REXEC、および SSH をセットアップするために必要なステップについては、「*Communications Server IP 構成ガイド*」(SC88-8926) に説明があります。詳細およびその他のセットアップ方法については、Developer for System z ライブラリー (<http://www-01.ibm.com/support/docview.wss?uid=swg27038517>) にあるホワイト・ペーパー「*Using INETD, REXEC and SSH with Rational Developer for System z*」(SC14-7301) を参照してください。

REXEC で使用される共通のポートは 512 です。使用されるポートを確認するには、`/etc/inetd.conf` および `/etc/services` を参照してください。

- `/etc/inetd.conf` で、`rexecd` サーバー (7 番目のワード) のサービス名 (最初のワード。この例では `exec`) を見つけます。

```
exec stream tcp nowait OMVSKERN /usr/sbin/orexecd rexecd -LV
```

- `/etc/services` で、そのサービス名 (最初のワード) に接続しているポート (2 番目のワード。この例では 512) を見つけます。

```
exec      512/tcp      #REXEC      Command Server
```

同じ原則が SSH にも適用されます。その共通のポートは 22 で、サーバー名は `sshd` です。

(オプション) インクルード・プリプロセッサのサポート

このカスタマイズ・タスクには、支援や特殊リソース、または特殊なカスタマイズ・タスクは必要ありません。

Developer for System z は、一部のサード・パーティーのインクルード・ステートメントを含む COBOL および PL/I インクルード・ステートメントの解釈と拡張をサポートします。Developer for System z では、サンプルの REXX `exec`、FEKRNPLI も提供され、これらは Developer for System z クライアントによって呼び出すことができ、PL/I コンパイラーを呼び出して PL/I ソースを拡張します。

FEKRNPLI は `FEK.#CUST.CNTL` に置かれます。ただし、`FEK.SFEKSAMP(FEKSETUP)` ジョブをカスタマイズして実行依頼したときに、別のロケーションを指定した場合は除きます。詳しくは、19 ページの『第 2 章 基本的なカスタマイズ』を参照してください。

サンプルの `FEK.#CUST.CNTL(FEKRNPLI) exec` を、このメンバーの中で説明されているようにカスタマイズします。以下の情報を指定する必要があります。

- `compiler_hlq`: PL/I コンパイラーの高位修飾子

Developer for System z クライアントは、TSO コマンド・サービスを使用して `exec` を実行します。このことは、FEKRNPLI `exec` が TSO コマンド・サービスの `SYSPROC`

または SYSEXEC 連結の中に置かれている場合、ユーザーは exec の正確なロケーションを知っている必要がないことを意味します。ユーザーが知っている必要があるのは名前だけです。デフォルトで、TSO コマンド・サービスは ISPF クライアント・ゲートウェイを使用して TSO 環境を作成しますが、ホワイト・ペーパー「Using APPC to provide TSO command services」(SC14-7291) に記載されているように APPC もサポートされています。ISPF クライアント・ゲートウェイを使用する場合、SYSPROC または SYSEXEC 連結は ISPF.conf で定義されます。このファイルのカスタマイズについて詳しくは、67 ページの『ISPF.conf、ISPF の TSO/ISPF クライアント・ゲートウェイ構成ファイル』を参照してください。

(オプション) Enterprise COBOL および PL/I での xUnit サポート

このカスタマイズ・タスクには、支援は必要ありませんが、以下のリソースまたは特殊なカスタマイズ・タスクが必要となります。

- LINKLIST 更新
-

反復可能な自己検査単体テストを実行するためのコードを作成する上で開発者を支援するフレームワークは、まとめて xUnit と呼ばれます。Developer for System z は、zUnit という、Enterprise COBOL および PL/I コードの単体テストのためのフレームワークを備えています。

zUnit フレームワークを使用するには、開発者は、STEPLIB または LINKLIST を通じて、FEK.SFEKLOAD ロード・ライブラリー内の AZU* および IAZU* ロード・モジュールにアクセスする必要があります。同様に、zUnit テスト実行機能の AZUTSTRN は、STEPLIB または LINKLIST のいずれかを通して、次のようにさまざまなシステム・ライブラリーにアクセスする必要があります。

- CEE.SCEERUN および CEE.SCEERUN2 (LE ランタイム)
- SYS1.CSSLIB (呼び出し可能システム・サービス)
- SYS1.SIXMLOD1 (XML ツールキット)

zUnit テスト実行機能は別のテスト・ケースを保持するロード・ライブラリーにもアクセスする必要があります。このライブラリーは大抵、開発者に固有のものです。

zUnit テスト実行機能の AZUTSTRN は、Developer for System z クライアントによってバッチ・モードで、TSO コマンド行から、および z/OS UNIX コマンド行から呼び出すことができます。

- Developer for System z は、バッチ・モードでの zUnit テスト実行機能の呼び出しを簡素化するためにサンプル・プロシージャ AZUZUNIT を提供しています。AZUZUNIT は FEK.#CUST.PROCLIB に置かれます。ただし、FEK.SFEKSAMP (FEKSETUP) ジョブをカスタマイズして実行依頼したときに、別のロケーションを指定した場合は除きます。詳しくは、19 ページの『第 2 章 基本的なカスタマイズ』を参照してください。

サンプル・プロシージャ FEK.#CUST.PROCLIB(AZUZUNIT) を、このメンバー内で説明されているようにカスタマイズし、SYS1.PROCLIB にコピーしてください。

プロシージャの名前とプロシージャ内のステップの名前は、Developer for System z クライアントに付属しているデフォルトのプロパティに一致していま

す。プロシージャーの名前またはプロシージャー内のステップの名前を変更する場合は、すべてのクライアント上の対応するプロパティ・ファイルを更新する必要があります。プロシージャー名とステップ名は変更しないでください。

AZUZUNIT プロシージャーをシステム・プロシージャー・ライブラリーにコピーできない場合は、呼び出しジョブに JCLLIB カードを JOB カードの直後に追加するように Developer for System z ユーザーに依頼します。

```
//MYJOB    JOB <job parameters>
//PROCS    JCLLIB ORDER=(FEK.#CUST.PROCLIB)
```

- z/OS UNIX から (/usr/lpp/rdz/bin/zunit スクリプトを使用して) zUnit テスト実行機能呼び出す場合、rsed.envvars の STEPLIB ディレクティブで、必要な非 LINKLIST データ・セットを指定して、開発者のセットアップ作業を簡素化することができます。

rsed.envvars は /etc/rdz/ に置かれます。ただし、FEK.SFEKSAMP(FEKSETUP) ジョブをカスタマイズして実行依頼したときに、別のロケーションを指定した場合は除きます。詳しくは、20 ページの『カスタマイズのセットアップ』を参照してください。このファイルは、TSO OEDIT コマンドで編集できます。

zunit スクリプトを使用すると、ユーザーはそのスクリプトによって使用される STEPLIB ディレクティブに追加するデータ・セットを指定できます。

- FEK.SFEKPROC(FEKZUNIT) exec を使用して TSO コマンド行から zUnit テスト・ランナーを呼び出す場合、システム・ライブラリーが LINKLIST に存在していなければなりません。存在しない場合、開発者は zUnit テスト・ランナーを呼び出すたびに、システム・データ・セット名を指定する必要があります。開発者のために、このデータ・セットの TSOLIB 割り振りを行うラッパー exec を作成することもできます。このラッパー exec のコーディング方法の例として、FEKZUNIT だけで使用することができます。

zUnit テスト・ランナーは、テスト・レポートの自動再フォーマット設定を可能にします。Developer for System z には、サンプルの変換 (Ant または junit フォーマットへの変換など) が用意されており、/usr/lpp/rdz/samples/zunit/xsd および /usr/lpp/rdz/samples/zunit/xsl に置かれています (Developer for System z をデフォルト・ロケーション /usr/lpp/rdz にインストールした場合)。

(オプション) エンタープライズ・サービス・ツール・サポート

このカスタマイズ・タスクには、支援や特殊リソース、または特殊なカスタマイズ・タスクは必要ありません。

Developer for System z クライアントには、エンタープライズ・サービス・ツールと呼ばれるコード生成コンポーネントがあります。生成されるコードのタイプによっては、このコードは Developer for System z ホスト・システム・インストールで提供される機能に依存しています。これらのホスト・システム機能を使用可能にする方法を、以下のセクションで説明しています。

- 117 ページの『第 5 章 (オプション) Application Deployment Manager (非推奨)』
- 142 ページの『(オプション) CICS 双方向言語サポート』

- 143 ページの『(オプション) 生成コードの診断用 IRZ メッセージ』

注: エンタープライズ・サービス・ツールには、サービス・フロー・モデラー (SFM) やエンタープライズ用 XML サービスなどの複数のツールが含まれます。

(オプション) CICS 双方向言語サポート

このカスタマイズ・タスクを完了するには、CICS 管理者の支援が必要になります。このタスクには、以下のリソースまたは特殊なカスタマイズ・タスクが必要です。

- CICS 領域 JCL を更新する
 - CICS に対してプログラムを定義する
-

Developer for System z エンタープライズ・サービス・ツールのコンポーネントは、さまざまな形式のアラビア語およびヘブライ語のインターフェース・メッセージをサポートしているほか、すべてのエディターおよびビューで双方向言語データの表示と編集をサポートしています。端末アプリケーションでは、左から右と、右から左の両方の画面がサポートされ、数値フィールド、および画面とは反対の向きのフィールドもサポートされます。

そのほかの双方向言語フィーチャーおよび機能には、以下のものがあります。

- エンタープライズ・サービス・ツールのサービス・リクエストは、インターフェース・メッセージの双方向属性を動的に指定します。
- サービス・フロー内の双方向データ処理は、双方向属性 (テキスト・タイプ、テキスト方向、数値スワッピング、対称スワッピングなど) に基づきます。これらの属性は、インターフェース・フローと端末フローのどちらの場合でも、さまざまなフロー作成のステージで指定できます。
- エンタープライズ・サービス・ツールの生成ランタイム・コードには、異なる双方向属性を持つメッセージ内のフィールド間でのデータの変換が含まれます。

さらに、エンタープライズ・サービス・ツールの生成コードは、CICS SFR (サービス・フロー・ランタイム) 以外の環境での bidi 変換をサポートできます。その一例がバッチ・アプリケーションです。エンタープライズ・サービス・ツールの生成ウィザードで適切な bidi 変換オプションを指定し、生成されたプログラムを適切な双方向変換ライブラリー FEK.SFEKLOAD とリンクすることにより、エンタープライズ・サービス・ツールの生成プログラムに、双方向変換ルーチンの呼び出しを組み込ませることができます。

CICS 双方向言語サポートをアクティブにするには、以下のタスクを実行します。

1. FEK.SFEKLOAD ロード・モジュール FEJBDCMP および FEJBDRX を CICS RPL 連結 (DD ステートメント DFHRPL) の中に置きます。この作業は、適用された保守が自動的に CICS で使用可能になるように、インストール・データ・セットを連結に追加する方法で行ってください。

重要: インストール・データ・セットを連結せず、モジュールを新規または既存のデータ・セットにコピーする場合は、このモジュールが DLL であり、PDSE ライブラリー内に存在しなければならないことに留意してください。

2. 適切な CEDA コマンドを使用して、FEJBDCMP および FEJBDTRX をプログラムとして CICS に対して定義します。

```
CEDA DEF PROG(FEJBDCMP) LANG(LE) G(xxx)
CEDA DEF PROG(FEJBDTRX) LANG(LE) G(xxx)
```

(オプション) 生成コードの診断用 IRZ メッセージ

このカスタマイズ・タスクには、支援は必要ありませんが、以下のリソースまたは特殊なカスタマイズ・タスクが必要となります。

- LINKLIST 更新
 - CICS 領域 JCL 更新
-

Developer for System z クライアントには、エンタープライズ・サービス・ツールと呼ばれるコード生成コンポーネントがあります。エンタープライズ・サービス・ツールを使用して生成したコードから診断用エラー・メッセージを出すためには、FEK.SFEKLMOD ロード・ライブラリー内のすべての IRZM* モジュールおよび IIRZ* モジュールを、その生成コードで使用できるようにする必要があります。エンタープライズ・サービス・ツールでは、以下の環境向けにコードを生成できます。

- CICS
- IMS
- MVS バッチ

生成コードが CICS トランザクションで実行される場合は、FEK.SFEKLMOD 内のすべての IRZM* モジュールと IIRZ* モジュールを CICS 領域の DFHRPL DD に追加します。この作業は、適用された保守が自動的に使用可能になるように、インストール・データ・セットを連結に追加する方法で行ってください。

それ以外の状態では、STEPLIB または LINKLIST を通じて、FEK.SFEKLMOD 内のすべての IRZM* モジュールと IIRZ* モジュールを使用可能にします。この作業は、適用された保守が自動的に使用可能になるように、インストール・データ・セットを連結に追加する方法で行ってください。

STEPLIB を使用する場合は、LINKLIST を介して使用できないモジュールを、コードを実行するタスクの STEPLIB ディレクティブで定義します。

ロード・モジュールを使用できず、生成コードでエラーが発生した場合は、次のメッセージが出されます。

```
IRZ9999S Failed to retrieve the text of a Language Environment runtime
message. Check that the Language Environment runtime message module for
facility IRZ is installed in DFHRPL or STEPLIB.
```

注:

- バージョン 8.5 では、IRZ* および IIRZ* ロード・モジュールと診断用メッセージは FEK.SFEKLOAD から FEK.SFEKLMOD ロード・ライブラリーに移動しています。

- モジュール FEK.SFEKLMOD(IRZPWSIO) は、トップダウン IMS MPP コード生成中に静的にリンクされます。したがって、このモジュールは、生成されたコードの実行時に使用可能であってはなりません。使用可能にできるのはコンパイル中のみです。
- バージョン 9.0.1 では、FEK.SFEKLMOD(IRZPWSIO) および関連する FEK.SFEKSAMP(IRZPWSH) サンプル PL/I インクルード・メンバーが Developer for System z から IMS バージョン 12 に移動されました。これらのパーツは、それぞれ IMS.SDFSRESL(DFSPWSIO) および IMS.SDFSAMPL(DFSPWSH) に名前変更されています。

(オプション) 統合デバッガー

このカスタマイズ・タスクを完了するには、セキュリティ、TCP/IP、および CICS の管理者の支援が必要になります。このタスクでは、以下のリソースまたは特殊なカスタマイズ・タスクが必要となります。

- SVC の追加 (IPL が必要)
 - SVC 用に更新された LPA
 - LINKLIST 更新
 - APF 許可
 - 開始タスクの定義
 - セキュリティ・プロファイルおよびアクセス・リストの定義
 - クライアント/ホスト通信およびホスト限定通信のための TCP/IP ポートの予約
 - (オプション) CICS 領域 JCL の更新
 - (オプション) CICS CSD の更新
 - (オプション) IMS メッセージ領域 JCL の更新
 - (オプション) DB2 ストアード・プロシージャのアドレス・スペース JCL の更新
 - (オプション) ELAXF* リモート・ビルド・プロシージャの更新
-

Developer for System z 統合デバッガー・ホスト・コンポーネントを使用すると、バージョン 9.0.1 以上のクライアントは、言語環境プログラム (Language Environment (LE)) ベースのさまざまなアプリケーションをデバッグできます。統合デバッガーには、z/OS 1.10 以上が必要です。

統合デバッガーのデータ・フローの概要については、「ホスト構成リファレンス」(SA88-4226-03) の『Developer for System z について』の章の『統合デバッガー』のセクションを参照してください。

サイトで統合デバッガーの使用を開始するには、以下のタスクを実行する必要があります。特に断りがない限り、すべてのタスクは必須です。

1. 統合デバッガーが機能するには、(必須の RSED および FEKD 開始タスクがアクティブになっている必要があります。DBGMR 始動 JCL については、30 ページの『DBGMR、デバッグ・マネージャー開始タスク』を参照してください。
2. 統合デバッガー構成は、DBGMR 開始タスクの始動引数で管理されます。詳しくは、147 ページの『統合デバッガー構成パラメーター』を参照してください。

3. DBGMGR 開始タスクを実行するには、FEK.SFEKAUTH ライブラリーに APF 許可がある必要があります。147 ページの『統合デバッガーの parmlib 更新』を参照してください。
4. 統合デバッガーは、Developer for System z 監視プログラム呼び出し (SVC) に依存しています。SVC ロード・モジュールは、IPL 時にリンク・バック域 (LPA) にロードされる必要があります。詳しくは、147 ページの『統合デバッガーの parmlib 更新』を参照してください。
5. 統合デバッガーには、アプリケーションからアクセスできる必要があり、STEPLIB 更新または LINKLIST 更新が必要です。詳しくは、147 ページの『統合デバッガーの parmlib 更新』を参照してください。
6. 統合デバッガーが機能するには、デバッグするアプリケーションのユーザー ID が有効な OMVS セグメントを持っている必要があります。詳しくは、148 ページの『統合デバッガーのセキュリティ更新』を参照してください。
7. 統合デバッガーは、いくつかのセキュリティ許可を必要とします。詳しくは、148 ページの『統合デバッガーのセキュリティ更新』を参照してください。
8. DBGMGR 開始タスクは、いくつかのセキュリティ許可を必要とします。詳しくは、148 ページの『統合デバッガーのセキュリティ更新』を参照してください。

CICS トランザクションをデバッグするために必要なステップは、以下のとおりです。

1. 統合デバッガーを使用すると、CICS トランザクションをデバッグできます。そのためには、統合デバッガーを CICS に定義する必要があります。詳しくは、150 ページの『統合デバッガーの CICS アップデート』を参照してください。

統合デバッガーは、特定のモジュールのデータ・セットから TEST ランタイム・オプションを読み取る、言語環境プログラム (Language Environment (LE)) のユーザー出口を使用できます。これは、IMS トランザクションや DB2 ストアード・プロシージャなど、サブシステムでアクティブになっているコードのデバッグで効果的です。このコードに動的に TEST ランタイム・オプションを提供することはできないためです。以下のステップは、LE ユーザー出口を使用している場合にのみ必要です。

1. LE ユーザー出口と、そのユーザー出口を呼び出すカスタマイズ済み LE 環境を作成します。詳しくは、149 ページの『統合デバッガーの言語環境プログラム (Language Environment) の更新』を参照してください。
2. カスタマイズされた LE 環境を IMS トランザクションに定義します。詳しくは、151 ページの『統合デバッガーの IMS の更新』を参照してください。
3. カスタマイズされた LE 環境を DB2 ストアード・プロシージャに定義します。詳しくは、151 ページの『統合デバッガーの DB2 ストアード・プロシージャの更新』を参照してください。

注:

- 統合デバッガーは、CICS CADP を使用して、CICS トランザクションに対する TEST ランタイム・オプションを提供します。CADP について詳しくは、CICS TS の資料を参照してください。ただし、必要に応じて、LE ユーザー出口メカニズムを使用することもできます。

- LE ユーザー出口メカニズムは、標準アプリケーションで使用することもできます。詳細については、151 ページの『統合デバッガーの ELAXF* の更新』を参照してください。

統合デバッガーと COBOL v4

COBOL v4 で作成されたプログラムをデバッグするために、統合デバッガーには、リストされているデータ・セット (PDS または PDS/E) へのアクセス権限が必要です。データ・セット名は、環境変数 AQE_DBG_V4LIST、または DD AQEV4LST を介して指定できます。どちらも存在しない場合、統合デバッガーは、実行可能ファイルのデータ・セットの最後の修飾子 (例、.LOAD) を .LISTING に置換することによって、データ・セット名を作成します。ご使用のサイトでどちらの方法が使用可能であるかを調べるには、開発者にご相談ください。

統合デバッガーおよびその他の言語環境プログラム・ベースのデバッガー

通常、所定のアプリケーション、CICS 領域、DB2 ストアード・プロシージャ、または IMS トランザクションでアクティブにできる言語環境プログラム (LE) ベースのデバッガー (統合デバッガーなど) は、1 つだけです。LE ベースのデバッガーであることを示す分かりやすい指標は、アプリケーションで使用されるべき CEEEVDBG ロード・モジュールまたは別名をそのデバッガーが提供することです。

ただし、バージョン 9.1.1 以上の統合デバッガーは、アプリケーションによって統合デバッガーが最初にロードされていれば、IBM Debug Tool for z/OS と共存させることができます。

統合デバッガーと暗号化通信

Developer for System z クライアントで SSL/TLS 暗号化を使用して RSE デーモンと通信している場合、(クライアント・ベースの) デバッグ・エンジンも、デフォルトで、(ホスト・ベースの) デバッグ・マネージャーと通信する際に暗号化を使用します。デフォルトでは、デバッグ・エンジンは Developer for System z クライアントと同じ証明書を使用します。

これは、RSE デーモンの開始タスクと、デバッグ・マネージャーの開始タスクが、通信の暗号化に関して同様にセットアップされていると想定されるということです。以下の代替シナリオは、RSE デーモンとデバッグ・マネージャーの暗号化設定が異なる状況において、有効です。

- 暗号化セッションのセットアップが失敗した場合、デバッグ・エンジンは、確認後に非暗号化通信の使用を試みることができます。
- デバッグ・エンジンは、Developer for System z クライアント以外の認証証明書を使用できます。

RSE デーモンとは異なり、デバッグ・マネージャーでは SSL/TLS 暗号化はネイティブ・サポートされていません。デバッグ・マネージャーは、暗号化通信について、Application Transparent Transport Layer Security (AT-TLS) と呼ばれる TCP/IP サービスに依存しています。段階的なセットアップ・ガイドについては、『IBM Rational Developer for System z ホスト構成リファレンス』(SA88-4226) の『AT-TLS のセットアップ』を参照してください。

統合デバッガー構成パラメーター

統合デバッガーでは、DBGMR 始動 JCL で以下の変数を構成できます。DBGMR 始動 JCL については、30 ページの『DBGMR、デバッグ・マネージャー開始タスク』を参照してください。

- タイムゾーン・オフセット (デフォルトは EST5DST)
- 外部 (クライアント/ホスト) 通信に使用されるポート番号 (デフォルトは 5335)
- 内部 (ホスト限定) 通信に使用されるポート (デフォルトは 5336)
- ロード・ライブラリーの高位修飾子 (デフォルトは FEK)

統合デバッガーの parmlib 更新

- DBGMR 開始タスクは、Developer for System z クライアントがホストに接続される前にアクティブになっている必要があります。IPL 時に自動的にサーバーを開始する方法についての詳細は、23 ページの『COMMNDxx への開始タスクの追加』を参照してください。
- DBGMR 開始タスクを実行するには、FEK.SFEKAUTH ライブラリーに APF 許可がある必要があります。詳しくは、25 ページの『PROGxx での APF 許可』を参照してください。
- 言語環境プログラム (LE) は、統合デバッガーを起動する必要があります。そのため、FEK.SFEKAUTH ライブラリーを LINKLIST に配置するか、またはデバッグするアプリケーションの STEPLIB に配置する必要があります。詳しくは、25 ページの『PROGxx での LINKLIST 定義』を参照してください。

注:

- LINKLIST を使用するときには、CEEVDBG ロード・モジュールを保持する他の LE ベース・デバッガーのライブラリーの前に FEK.SFEKAUTH があることを確認してください。例えば、IBM Debug Tool for z/OS は hlq.SEQA* ライブラリーを使用します。
- 競合を回避するには、LINKLIST で定義される LE ベースのデバッガーは 1 つでなければなりません。
- 統合デバッガーは、アプリケーションによって統合デバッガーが最初にロードされていれば、IBM Debug Tool for z/OS と共存させることができます。
- FEK.SFEKAUTH 内の統合デバッガー・ロード・モジュールは、デバッグ・マネージャー開始タスクを除いて、許可済みで実行する必要はありません。ここには、ロード・モジュールが常駐しているので、それらを許可済みの環境で使用できます。
- 統合デバッガーは z/OS Binder を使用します。これは、SYS1.MIGLIB が LINKLIST (または STEPLIB) になければならないことを意味します。詳しくは、27 ページの『必要な LINKLIST 定義と LPA 定義』を参照してください。
- 統合デバッガーは z/OS Binder API を使用します。この API は z/OS 1.10 以降で /usr/lib/iewbndd.so として使用でき、z/OS 1.13 以降でも SYS1.SIEAMIGE(IEWBNDD) として使用できます。これは、z/OS 1.13 以上では、SYS1.SIEAMIGE が LINKLIST (または STEPLIB) 内にある必要があることを示しています。詳しくは、27 ページの『必要な LINKLIST 定義と LPA 定義』を参照してください。

注: z/OS 1.13 以上のシステムで SYS1.SIEAMIGE が LINKLIST または STEPLIB に含まれていない場合、統合デバッガーは以下のメッセージを発行し、/usr/lib/iewbndd.so の使用を試みます。

CEE3501S The module //IEWBNDD was not found

- 統合デバッガーは、システムに Developer for System z 監視プログラム呼び出し (SVC) が定義されていることを必要とします。デフォルトの SVC 番号は 251 です。関連するロード・モジュール FEK.SFEKLPA(AQESVC03) が、IPL 時に LPA にロードされる必要があります。詳しくは、23 ページの『IEASVCxx での SVC 定義』、および 24 ページの『LPALSTxx での LPA 定義』を参照してください。

注: バージョン 9.1.1 より前のバージョンの Developer for System z では、現在使用されていない別の SVC、AQESVC01 を使用していました。これは、ご使用のホスト・システムにバージョン 9.1.1 以上の Developer for System z しかない場合には、削除できます。

統合デバッガーの TCP/IP 更新

統合デバッガーは 2 つの TCP/IP ポートを使用します。詳細については、「ホスト構成リファレンス」(SA88-4226) の『TCP/IP に関する考慮事項』を参照してください。

- クライアント/ホスト通信用のポート (デフォルトは 5335)。このポートでの通信を暗号化することができます。
- ホスト限定の通信用のポート (デフォルトは 5336)。

注: 統合デバッガーでは、IPv6 通信はサポートされません。

統合デバッガーのセキュリティ更新

統合デバッガーは以下のセキュリティ定義を必要とします。詳しくは、171 ページの『第 9 章 セキュリティ定義』を参照してください。

- デバッグ対象のアプリケーションを実行しているユーザー ID の OMVS セグメント (これには、CICS 領域のユーザー ID が含まれます)
- DBGMGR 開始タスク
- 開始タスク・ユーザー ID の BPX.SERVER 許可
- 開始タスク・ロード・ライブラリーのプログラム制御
- 問題プログラム状態をデバッグしているユーザーの AQE.AUTHDEBUG.STDPGM 許可
- 問題プログラム状態の許可済みプログラムをデバッグしているユーザーの AQE.AUTHDEBUG.AUTHPGM 許可

注: 統合デバッガーがない状態の、既存の Developer for System z セットアップからのマイグレーションを単純化するために、RACF コマンドを伴うサンプル JCL FEK.SFEKSAMP(AQERACF) が提供されています。これは統合デバッガーに関連するセキュリティ定義のみを定義します。

統合デバッガーの言語環境プログラム (Language Environment) の更新

統合デバッガーは、特定のモジュールのデータ・セットから TEST ランタイム・オプションを読み取る、言語環境プログラム (Language Environment (LE)) のユーザー出口を使用できます。これは、IMS トランザクションや DB2 ストアード・プロシージャなど、サブシステムでアクティブになっているコードのデバッグで効果的です。このコードに動的に TEST ランタイム・オプションを提供することはできないためです。

このプロセス・フローについて、次のステップで説明します。

1. LE がモジュールの実行を依頼されます。
2. LE は、引数としてモジュール名を指定してユーザー出口を呼び出します。
3. ユーザー出口は、TEST ランタイム・オプション・データ・セットを開き、モジュールの TEST ランタイム・オプションを見つけます。
4. ユーザー出口は、LE に TEST ランタイム・オプションを返します。
5. LE は、指定された TEST ランタイム・オプションを使ってモジュールを実行します。

注:

- IBM Debug Tool for z/OS は、ここで説明するユーザー出口メカニズムの拡張バージョンを使用します。共存をサポートし、TEST ランタイム・オプションを保守する労力が重複することを回避するため、統合デバッガーは、ユーザー出口オプションおよび TEST ランタイム・オプションのデータ・セット名について、デバッグ・ツールと同じデフォルトを使用します。
- ユーザー出口では、TEST ランタイム・オプションのデータ・セット名で、ユーザー ID およびモジュール名に対する変数置換をサポートしています。
- Developer for System z クライアント GUI を使用して TEST ランタイム・オプションのデータ・セットを保守するには、Developer for System z バージョン 9.1.0 以上のホストおよびクライアントが必要です。
- 統合デバッガーは、CICS CADP を使用して、CICS トランザクションに対する TEST ランタイム・オプションを提供します。CADP について詳しくは、CICS TS の資料を参照してください。

以下のステップは、LE ユーザー出口を使用している場合にのみ必要です。

- AQED3CXT をカスタマイズおよび実行依頼して、ユーザー出口を作成します。AQED3CXT は FEK.#CUST.JCL にあります。ただし、FEK.SFEKSAMP(FEKSETUP) ジョブをカスタマイズして実行依頼したときに別のロケーションを指定した場合は除きます。
- AQED3CEE をカスタマイズおよび実行依頼して、ユーザー出口を呼び出すカスタマイズ済み LE ロード・モジュールを作成します。AQED3CEE は FEK.#CUST.JCL にあります。ただし、FEK.SFEKSAMP(FEKSETUP) ジョブをカスタマイズして実行依頼したときに別のロケーションを指定した場合は除きます。デフォルトでは、カスタマイズした LE ロード・モジュール (CEE*) は FEK.#CUST.LOAD に置かれています。

注: パフォーマンス上の理由から、カスタマイズした LE ロード・モジュールが入っているロード・ライブラリーを LINKLIST には配置しないでください。購入したソフトウェアも含めて、すべての LE 対応ロード・モジュールに対してユーザー出口が呼び出されることになります。

統合デバッガーの CICS アップデート

統合デバッガーで CICS トランザクションをデバッグするには、以下の CICS アップデートが必要です。

- CICS システム初期設定 (SIT) パラメーターの更新:
 - DEBUGTOOL=YES を指定します。
 - TCPIP=YES を指定します。
 - LINKLIST によって、DFHRPL DD 連結からロード・モジュールを取り出す場合、LLACOPY=YES を指定します。
 - ユーザーが統合デバッガーの SVC (読み取り専用メモリーにロードされたトランザクションをデバッグするために必要) を使用することを許可しない場合、RENTPGM=NOPROTECT を指定します。
- CICS JCL 更新:
 - EXEC ステートメントで REGION=0M を指定します。
 - 領域の DFHRPL DD ステートメントで FEK.SFEKAUTH ロード・ライブラリーを定義します。SIT パラメーター LLACOPY=YES が指定されている場合、ライブラリーは LINKLIST に置くこともできます。
 - 領域の DFHRPL DD ステートメントで SYS1.MIGLIB ロード・ライブラリーを定義します。SIT パラメーター LLACOPY=YES が指定されている場合、ライブラリーは LINKLIST に置くこともできます。
 - z/OS 1.13 以上の場合、領域の DFHRPL DD ステートメントで SYS1.SIEAMIGE ロード・ライブラリーを定義します。SIT パラメーター LLACOPY=YES が指定されている場合、ライブラリーは LINKLIST に置くこともできます。詳細については、147 ページの『統合デバッガーの parmlib 更新』にある z/OS Binder API の情報を参照してください。

注:

- CICS 領域のユーザー ID には、SIT パラメーター LLACOPY=YES が意図したとおりに機能するために、FACILITY クラスの CSVLLA.dataset プロファイルに対する UPDATE 許可が必要です。
- COBOL v4 で作成されたプログラムをデバッグするために、統合デバッガーには、リストされているデータ・セット (PDS または PDS/E) へのアクセス権限が必要です。データ・セット名は、環境変数 AQE_DBG_V4LIST、または DD AQEV4LST を介して指定できます。どちらも存在しない場合、統合デバッガーは、実行可能ファイルのデータ・セットの最後の修飾子 (例、.LOAD) を .LISTING に置換することによって、データ・セット名を作成します。ご使用のサイトでどちらの方法が使用可能であるか、開発者に問い合わせてください。
- CICS CSD アップデート:

AQECSD サンプル CSD 更新ジョブの説明に従って、デバッガーを CICS 領域に定義します。AQECSD は FEK.#CUST.JCL にあります。ただし、ジョブ FEK.SFEKSAMP(FEKSETUP) をカスタマイズして実行依頼したときに別のロケーションを指定した場合を除きます。詳細については、20 ページの『カスタマイズのセットアップ』を参照してください。

統合デバッガーの IMS の更新

統合デバッガーで IMS トランザクションをデバッグするには、以下の IMS の更新が必要です。

- IMS メッセージ領域 (DFSMPR) JCL の更新:
 - 領域の EXEC ステートメントで REGION=0M を指定します。
 - FEK.SFEKAUTH ロード・ライブラリーが LINKLIST に存在しない場合、このライブラリーを STEPLIB に追加します。統合デバッガー以外の他の製品で必要とされている場合を除いて、すべての STEPLIST データ・セットで APF 許可を保守する必要はありません。
 - 言語環境プログラム (Language Environment (LE)) ユーザー出口を使用している場合、カスタマイズされた LE ロード・モジュールが入っているロード・ライブラリー (デフォルトでは FEK.#CUST.LOAD) を STEPLIB に追加します。

統合デバッガーの DB2 ストアード・プロシージャの更新

統合デバッガーで DB2 ストアード・プロシージャをデバッグするには、以下の DB2 関連の更新を行う必要があります。

- DB2 ストアード・プロシージャのアドレス・スペースの開始タスク JCL に対する更新
 - EXEC ステートメントで REGION=0M を指定します。
 - FEK.SFEKAUTH ロード・ライブラリーが LINKLIST に存在しない場合、このライブラリーを STEPLIB に追加します。統合デバッガー以外の他の製品で必要とされている場合を除いて、すべての STEPLIST データ・セットで APF 許可を保守する必要はありません。
 - 言語環境プログラム (Language Environment (LE)) ユーザー出口を使用している場合、カスタマイズされた LE ロード・モジュールが入っているロード・ライブラリー (デフォルトでは FEK.#CUST.LOAD) を STEPLIB に追加します。

統合デバッガーの ELAXF* の更新

以下の更新は、Developer for System z の提供するリモート・ビルド・プロシージャで統合デバッガーを使用するために必要です。これらの JCL プロシージャの機能および場所について詳しくは、35 ページの『ELAXF* リモート・ビルド・プロシージャ』を参照してください。

- ELAXFGO JCL プロシージャに対する更新:
 - EXEC ステートメントで REGION=0M を指定します。
 - FEK.SFEKAUTH ロード・ライブラリーが LINKLIST に存在しない場合、このライブラリーを STEPLIB に追加します。統合デバッガー以外の他の製品で必要とされている場合を除いて、すべての STEPLIST データ・セットで APF 許可を保守する必要はありません。

- 言語環境プログラム (LE) ユーザー出口を使用している場合、カスタマイズされた LE ロード・モジュールが入っているロード・ライブラリー (デフォルトでは FEK.#CUST.LOAD) を STEPLIB に追加します。

(オプション) 問題判別ツールのサポート

このカスタマイズ・タスクには、支援や特殊リソース、または特殊なカスタマイズ・タスクは必要ありません。

Developer for System z は、さまざまな IBM z/OS 問題判別ツールと統合することができます。以下のセクションで、これらのツールを Developer for System z クライアントで使用できる方法について説明します。

- IBM Debug Tool for z/OS: 『(オプション) DB2 および IMS のデバッグのサポート』を参照してください。バージョン 9.0.1 以降では、Developer for System z に統合デバッガーが備わっていることに注意してください。これを Debug Tool の代わりに使用できます。
- IBM File Manager for z/OS: 153 ページの 『(オプション) File Manager のサポート』を参照してください。
- IBM Fault Analyzer for z/OS: Developer for System z ホスト・システム構成は必要ありません。バージョン 9.0 以降、Developer for System z で Fault Analyzer 統合 (FAI) がサポートされなくなったことに注意してください。このサポートがまだ残っている古いクライアントでは、この機能をアンインストールし、IBM Fault Analyzer プラグイン (Eclipse 用) をインストールする必要があります。このプラグインは、『IBM Problem Determination Tools Plug-ins』の Web ページ (<http://www-01.ibm.com/software/awdtools/deployment/pdtpplugins/>) から入手できます。

(オプション) DB2 および IMS のデバッグのサポート

このカスタマイズ・タスクには、支援や特殊リソース、または Developer for System z 構成の特殊なカスタマイズ・タスクは必要ありません。ただし、IBM Debug Tool for z/OS 構成に関する要件があります。

IBM Debug Tool for z/OS には、カスタマイズされた言語環境プログラム (LE) ユーザー出口 (CEEEXITA) が用意されています。これは、IMS ストアード・プロシージャおよび DB2 ストアード・プロシージャの LE 初期設定ロジックによって呼び出された場合に、TEST ランタイム・オプションを返します。IBM Debug Tool for z/OS は、z/OS システム上で TEST ランタイム・オプションのデータ・セットを作成して管理するための、Problem Determination Tools Common Components サーバーに対する Debug Tool 拡張機能も提供します。Developer for System z は、IMS および DB2 ストアード・プロシージャのランタイム用のデバッグ・プロファイルを管理するために IBM Debug Tool for z/OS のサポートを使用して拡張することができます。

IBM Debug Tool for z/OS の資料で、必要なセットアップが詳しく説明されているため、ここでは簡単に説明します。

- 言語環境プログラムのユーザー出口 (hlq.SEQA*) での TEST ランタイム・オプションの指定
- 「DTSP プロファイル」ビューのサポートの追加
 - Problem Determination Tools Common Components サーバー (hlq.SIPV*、ジョブ IPVGSVRJ) のインストール
 - hlq.SEQA* 問題判別ツール共通コンポーネントの Debug Tool 機能拡張のインストールおよび構成

注:

- IBM Debug Tool for z/OS 製品は、別個に取得、インストール、および構成する必要があります。この製品のインストールとカスタマイズについては、本書には記載されていません。
- Developer for System z クライアントは、DTSP Profile view plug-in for Eclipse を使用しません。
- Developer for System z クライアントでは、通常のバッチ・モード・デバッグのために言語環境プログラムのユーザー出口を使用しません。
- Developer for System z クライアントは、問題判別ツール共通コンポーネント・サーバーと直接通信します。つまり、ユーザーはこのポート番号を知っている必要があります、このサーバーによって使用されるポートが、z/OS ホスト・システムを保護しているファイアウォールで開いている必要があります。

(オプション) File Manager のサポート

このカスタマイズ・タスクには、支援や特殊リソース、または Developer for System z 構成の特殊なカスタマイズ・タスクは必要ありません。ただし、IBM File Manager for z/OS 構成に関する要件があります。

Developer for System z の IBM File Manager for z/OS との初期統合は、Developer for System z バージョン 8.0.3 で非推奨となっており、バージョン 8.5 ではサポートされなくなっています。この機能によって提供されるサービスは、別のエリアに移動されています。不定形式 QSAM 編集をはじめとする一部の機能は、Developer for System z による標準データ・セット処理に組み込まれました。コピーブックまたはインクルード・ファイルを使用する定様式データ編集など、さらに上級の機能を使用するには、IBM File Manager plug-in for Eclipse が Developer for System z クライアントにインストールされている必要があります。このプラグインは、『IBM Problem Determination Tools Plug-ins』の Web ページ (<http://www-01.ibm.com/software/awdtools/deployment/pdtpplugins/>) から入手できます。

IBM File Manager プラグイン (Eclipse 用) は、問題判別ツール・サーバーを使用して File Manager サービスにアクセスします。このサーバーは、File Manager ISPF パネル・インターフェースでは使用されません。そのため、問題判別ツールに固有の、追加の File Manager セットアップ・タスクがあります。詳細については、File Manager の資料を参照してください。

問題判別ツール・サーバーで使用するポート番号は、rsed.envvars ディレクティブの PD_SERVER_PORT で指定する必要があります。

rsed.envvars は /etc/rdz/ に置かれます。ただし、FEK.SFEKSAMP(FEKSETUP) ジョブをカスタマイズして実行依頼したときに、別のロケーションを指定した場合は除きます。詳しくは、20 ページの『カスタマイズのセットアップ』を参照してください。このファイルは、TSO **OEDIT** コマンドで編集できます。

注:

- IBM File Manager for z/OS 製品は、別個に取得、インストール、および構成する必要があります。この製品のインストールとカスタマイズについては、本書には記載されていません。
- Developer for System z クライアントは、問題判別ツール・サーバーと直接通信します。つまり、このサーバーによって使用されるポートが、z/OS ホスト・システムを保護しているファイアウォールで開いている必要があります。

(オプション) WORKAREA と /tmp のクリーンアップ

このカスタマイズ・タスクには、支援や特殊リソース、または特殊なカスタマイズ・タスクは必要ありません。

ISPF の TSO/ISPF クライアント・ゲートウェイおよび SCLM Developer Toolkit 機能は WORKAREA および /tmp ディレクトリーに一時作業ファイルを保管しますが、それらのファイルは、セッションが閉じる前に除去されます。ところが、処理中に通信エラーが発生した場合など、一時出力が残される場合があります。そのため、WORKAREA および /tmp ディレクトリーを時々クリアしてください。

z/OS UNIX には、ファイルが入っているディレクトリーとファイルの経過日数にしたがってファイルを削除する、skulker というシェル・スクリプトが用意されています。指定された日時にコマンドを実行する z/OS UNIX cron デーモンと結合すれば、定期的にターゲット・ディレクトリーを空にする自動化ツールをセットアップできます。skulker スクリプトおよび cron デーモンの詳細については、「UNIX System Services コマンド解説書」(SA88-8641) を参照してください。

注: WORKAREA は /var/rdz/ に置かれます。ただし、FEK.SFEKSAMP(FEKSETUP) ジョブをカスタマイズして実行依頼したときに、別のロケーションを指定した場合は除きます。詳しくは、20 ページの『カスタマイズのセットアップ』を参照してください。

第 8 章 インストール検査

製品のカスタマイズの完了後、この章で説明するインストール検査プログラム (IVP) を使用して、主要な製品コンポーネントのセットアップが正常であることを検査できます。

開始タスクの検査

JMON、JES ジョブ・モニター

JMON 開始タスクまたはユーザー・ジョブを開始します。DD SYSOUT での開始情報が、次のメッセージで終わることを確認します。

```
FEJ211I Server ready to accept connections.
```

ジョブが戻りコード 66 を伴って終了する場合は、FEK.SFEKAUTH に APF 許可がありません。

注: 他の IVP テストを続行する前に、JES ジョブ・モニターを始動してください。

RSED、RSE デーモン

RSED 開始タスクまたはユーザー・ジョブを、IVP=IVP パラメーターを指定して開始します。このパラメーターを指定すると、サーバーはいくつかのインストール検査テストを行った後に終了します。それらのテストの出力は、DD STDOUT で入手できます。エラーが発生した場合は、DD STDERR でもデータを入手できます。DD STDOUT を確認して、以下の IVP が成功したことを示すメッセージの有無を調べます。

- Java の始動
- JES ジョブ・モニター接続
- TCP/IP のセットアップ

STDOUT データは、次のサンプルのようになります。

```
-----  
RSE daemon startup script  
-----
```

```
arguments: IVP -C/etc/rdz -P
```

```
RSE daemon IVP test
```

```
CDFMVS08 -- Fri Mar 23 17:50:52 2012 UTC  
uid=8(STCRSE) gid=1(STCGROUP)
```

```
started from /usr/lpp/rdz/bin/rsed.sh  
startup script version Aug09,2012
```

```
configuration files located in /etc/rdz -- startup argument  
daemon port is 4035 -- set in rsed.envvars  
debug level is 1 -- set in rsecomm.properties  
TMPDIR=/tmp -- default
```

```

-----
current environment variables
-----
@="/usr/lpp/rdz/bin/rsed.sh" @[1]="-C/etc/rdz" @[2]="-P"
ANT_HOME="/usr/lpp/Apache/Ant/apache-ant-1.7.1"
CGI_DTWORk="/var/rdz"
CGI_ISPCONF="/etc/rdz"
CGI_ISPHOME="/usr/lpp/ispf"
CGI_ISPWORK="/var/rdz"
CGI_TRANTABLE="FEK.#CUST.LSTRANS.FILE"
CLASSPATH=".:usr/lpp/rdz/lib:usr/lpp/rdz/lib/dstore_core.jar:usr/lpp/
ERRNO="0"
HOME="/tmp"
IFS="
"
JAVA_HOME="/usr/lpp/java/J6.0"
JAVA_PROPAGATE="NO"
LANG="C"
LIBPATH=".:usr/lib:usr/lpp/java/J6.0/bin:usr/lpp/java/J6.0/bin/classi
LINENO="66"
LOGNAME="STCRSE"
MAILCHECK="600"
OLDPWD="/tmp"
OPTIND="1"
PATH=".:usr/lpp/java/J6.0/bin:usr/lpp/rdz/bin:usr/lpp/ispf/bin:/bin:/
PPID="33554711"
PS1="¥$ "
PS2="> "
PS3="#? "
PS4="+ "
PWD="/etc/rdz"
RANDOM="27298"
RSE_CFG="/etc/rdz"
RSE_HOME="/usr/lpp/rdz"
RSE_LIB="/usr/lpp/rdz/lib"
SECONDS="0"
SHELL="/bin/sh"
STEPLIB="NONE"
TMPDIR="/tmp"
TZ="EST5EDT"
X_ARG="-T"
X_C="-- startup argument"
X_KEY="-T"
X_L="-- set in rsecomm.properties"
X_LOG="1"
X_P="-- set in rsed.envvars"
X_PORT="4035"
X_VAL=""
_="-----"
_BPX_SHAREAS="YES"
_BPX_SPAWN_SCRIPT="YES"
_CEE_DMPTARG="/tmp"
_CEE_RUNOPTS="ALL31(ON) HEAP(32M,32K,ANYWHERE,KEEP,,) TRAP(ON)"
_CMDSERV_BASE_HOME="/usr/lpp/ispf"
_CMDSERV_CONF_HOME="/etc/rdz"
_CMDSERV_WORK_HOME="/var/rdz"
_EDC_ADD_ERRNO2="1"
_RSE_ISPF_OPTS="&SESSION=SPAWN"
_RSE_DAEMON_CLASS="com.ibm.etools.zos.server.RseDaemon"
_RSE_DAEMON_IVP_TEST="1"
_RSE_HOST_CODEPAGE="IBM-1047"
_RSE_JAVA_OPTS=" -DISPF_OPTS='&SESSION=SPAWN' -DA_PLUGIN_PATH=
_RSE_JMON_PORT="6715"
_RSE_LOG_LEVEL="1"
_RSE_POOL_SERVER_CLASS="com.ibm.etools.zos.server.ThreadPoolProcess"
_RSE_RSED_PORT="4035"
_RSE_SAF_CLASS="/usr/include/java_classes/IRRRacf.jar"

```

```

_RSE_SCRIPT_VERSION="Jan09,2012"
_RSE_SERVER_CLASS="org.eclipse.dstore.core.server.Server"
_RSE_SERVER_TIMEOUT="120000"
_SCLMDT_BASE_HOME="/usr/lpp/rdz"
_SCLMDT_CONF_HOME="/var/rdz/scldmt"
_SCLMDT_TRANTABLE="FEK.#CUST.LSTRANS.FILE"
_SCLMDT_WORK_HOME="/var/rdz"
debug_level="1"

```

Address Space size limits

```

current address space size limit is 1913626624 (1825.0 MB)
maximum address space size limit is 2147483647 (2048.0 MB)

```

service history

```

Fri Jun 14 13:47:39 2013 -- COPY -- HHOP900 v9000 created 14 Jun 2013

```

java service level

```

java full version "J2RE 1.6.0 IBM z/OS build pmz3160sr13-20130207_01(SR13)

```

LE runtime options

```

Options Report for Enclave main 05/23/12 1:50:52 PM
Language Environment V01 R11.00

```

LAST WHERE SET	OPTION
Installation default	ABPERC(NONE)
Programmer default	ABTERMENC(RETCODE)
Installation default	NOAIXBLD
Invocation command	ALL31(ON)
Programmer default	ANYHEAP(32768,16384,ANYWHERE,FREE)
Installation default	NOAUTOTASK
Programmer default	BELOWHEAP(32768,16384,FREE)
Installation default	CBLOPTS(ON)
Installation default	CBLP SHPOP(ON)
Installation default	CBLQDA(OFF)
Installation default	CEEDUMP(60,SYSOUT=*,FREE=END,SPIN=UNALL
Installation default	CHECK(ON)
Installation default	COUNTRY(US)
Installation default	NODEBUG
Installation default	DEPTHCONDLMT(10)
Installation default	DYNDUMP(*USERID,NODYNAMIC,TDUMP)
Installation default	ENVAR("")
Installation default	ERRCOUNT(0)
Installation default	ERRUNIT(6)
Installation default	FILEHIST
Installation default	FILETAG(NOAUTOCVT,NOAUTOTAG)
Default setting	NOFLOW
Invocation command	HEAP(33554432,32768,ANYWHERE,KEEP,16384
Installation default	HEAPCHK(OFF,1,0,0,0)
Installation default	HEAPPOLS(OFF,8,10,32,10,128,10,256,10,
Installation default	INFMSGFILTER(OFF,,))
Installation default	INQPCOPN
Installation default	INTERRUPT(OFF)
Programmer default	LIBSTACK(32768,16384,FREE)
Installation default	MSGFILE(SYSOUT,FBA,121,0,NOENQ)
Installation default	MSGQ(15)
Installation default	NATLANG(ENU)
Ignored	NONONIPSTACK(See THREADSTACK)


```

Installation default      OCSTATUS
Installation default      NOPC
Installation default      PLITASKCOUNT(20)
Programmer default        POSIX(ON)
Installation default      PROFILE(OFF,"")
Installation default      PRTUNIT(6)
Installation default      PUNUNIT(7)
Installation default      RDRUNIT(5)
Installation default      RECPAD(OFF)
Invocation command        RPTOPTS(ON)
Installation default      RPTSTG(OFF)
Installation default      NORTEREUS
Installation default      NOSIMVRD
Programmer default
STACK(65536,65536,ANYWHERE,KEEP,524288,131072)
Installation default      STORAGE(NONE,NONE,NONE,0)
Installation default      TERMTHDACT(TRACE,,96)
Installation default      NOTEST(ALL,"*","PROMPT","INSPREF")
Installation default      THREADHEAP(4096,4096,ANYWHERE,KEEP)
Installation default
THREADSTACK(OFF,4096,4096,ANYWHERE,KEEP,131072,
Installation default      TRACE(OFF,4096,DUMP,LE=0)
Invocation command        TRAP(ON,SPIE)
Installation default      UPSI(00000000)
Installation default      NOUSRHDLR(,)
Installation default      VCTRSAVE(OFF)
Installation default      XPLINK(OFF)
Installation default      XUFLOW(AUTO)

```

```

-----
java startup test...
-----

```

```

java full version "JRE 1.6.0 IBM z/OS build pmz3160sr13-20130207_01
(SR13)"
java version "1.6.0"
Java(TM) SE Runtime Environment (build pmz3160sr13-20130207_01(SR13))
IBM J9 VM (build 2.4, JRE 1.6.0 IBM J9 2.4 z/OS s390-31 jvmmz3160sr13-
20130114_1
J9VM - 20130114_134867
JIT - r9_20130108_31100
GC - 20121212_AA
JCL - 20130204_01

```

```

-----
JES Job Monitor test...
-----

```

```

executed on CDFMVS08 -- Fri Mar 23 17:50:52 EDT 2012
executed by uid=8(STCRSE) gid=1(STCGROUP)
using /etc/rdz/rsed.envvars

```

```

current address space size limit is 1913626624 (1825.0 MB)
maximum address space size limit is 2147483647 (2048.0 MB)

```

```

testing JES Job Monitor on port 6715...
hostName=CDFMVS08
hostAddr=9.42.112.75
IPv4 is supported
Waiting for JES Job Monitor response...
ACKNOWLEDGE01v03
Success

```

```

-----
TCP/IP IVP test...
-----

```

```

executed on CDFMVS08 -- Fri Mar 23 17:50:53 EDT 2012

```

```

executed by uid=8(STCRSE) gid=1(STCGRP)
using /etc/rdz/rsed.envvars

current address space size limit is 1913626624 (1825.0 MB)
maximum address space size limit is 2147483647 (2048.0 MB)
-----
TCP/IP resolver configuration (z/OS UNIX search order):
-----
Resolver Trace Initialization Complete -> 2012/05/23 17:50:54.208378

res_init Resolver values:
Global Tcp/Ip Dataset = None
Default Tcp/Ip Dataset = None
Local Tcp/Ip Dataset = /etc/resolv.conf
Translation Table = Default
UserId/JobName = STCRSE
Caller API = LE C Sockets
Caller Mode = EBCDIC
(L) DataSetPrefix = TCPIP
(L) HostName = CDFMVS08
(L) TcpIpJobName = TCPIP
(L) DomainOrigin = RALEIGH.IBM.COM
(L) NameServer = 9.42.206.2
               9.42.206.3
(L) NsPortAddr = 53 (L) ResolverTimeout = 10
(L) ResolveVia = UDP (L) ResolverUdpRetries = 1
(*) Options NDots = 1
(*) SockNoTestStor
(*) AlwaysWto = NO (L) MessageCase = MIXED
(*) LookUp = DNS LOCAL
res_init Succeeded
res_init Started: 2012/05/23 17:50:54.229888
res_init Ended: 2012/05/23 17:50:54.229898
*****
MVS TCP/IP NETSTAT CS V1R11 TCPIP Name: TCPIP 17:50:54
Tcpi started at 11:31:40 on 05/23/2012 with IPv6 enabled

-----
host IP address:
-----
hostName=CDFMVS08
hostAddr=9.42.112.75
bindAddr=9.42.112.75
localAddr=9.42.112.75

Success, addresses match

-----
RSE daemon IVP ended -- return code 0 -- Fri Mar 23 17:50:55 EDT 2012
-----

注: 他の IVP テストを続行する前に、IVP パラメーターを指定せずに RSE デーモンを始動してください。始動が成功すると、RSE デーモンは次のコンソール・メッセージを発行します。

FEK002I RseDaemon started. (port=4035)

```

DBGMGR、デバッグ・マネージャー

オプションの DBGMGR 開始タスクまたはユーザー・ジョブを開始します。正常に開始されると、サーバーは以下のコンソール・メッセージを出します。clientport は外部 (クライアント/ホスト) 通信に使用されるポートの番号、hostport は内部 (ホスト限定) 通信に使用されるポート番号です。

```
AQECM001I Debug Manager startup complete (clientport/hostport)
```

ジョブが戻りコード 66 を伴って終了する場合は、FEK.SFEKAUTH に APF 許可がありません。

注: 続けて他のデバッグ関連 IVP テストを行う前に、DBGMGR を開始してください。

IVP オペレーター・コマンド

アクティブ RSE デーモンは **IVP** 変更コマンドをサポートするため、これを使用してコンソールから選択した IVP を実行することができます。

PassTicket の再使用可能性

Developer for System z では、PassTicket の生成が 1 秒ごとに 1 ユーザーにつき 1 つに限定されているため、生成する PassTickets が再利用可能でなければなりません。PassTicket の再使用可能性を確認するには、以下のオペレーター・コマンドを実行します。userid は、有効な TSO ユーザー ID に置き換えてください。

```
MODIFY RSED,APPL=IVP PASSTICKET,userid
```

このコマンドは、次のサンプルのような出力を返します。

```
MODIFY RSED,APPL=IVP PASSTICKET,IBMUSER
```

```
+FEK900I PASSTICKET IVP: start: serverid=STCRSE userid=IBMUSER
+FEK900I PASSTICKET IVP: the default applid=FEKAPPL
+FEK900I PASSTICKET IVP: Success, PassTicket IVP finished normally
+FEK901I PASSTICKET IVP Exit code = 0
```

RSE デーモン接続

次のコマンドを実行することにより、RSE デーモン接続を検査します。userid は、有効な TSO ユーザー ID に置き換えてください。

```
MODIFY RSED,APPL=IVP DAEMON,userid
```

このコマンドは、162 ページの『サービスの検査』で説明した fekfivpd IVP と機能的是に同じですが、パスワードが必要ないという利点があります。RSE は PassTicket を生成し、それをパスワードとして使用します。このコマンドは、次のサンプルのような出力を返します。

```
F RSED,APPL=IVP DAEMON,IBMUSER
```

```
+FEK900I DAEMON IVP: SSL is disabled
+FEK900I DAEMON IVP: connected
+FEK900I DAEMON IVP: 1343
+FEK900I DAEMON IVP: 8878350
+FEK900I DAEMON IVP: Success
+FEK901I DAEMON IVP Exit code = 0
```

ISPF クライアント・ゲートウェイ

次のコマンドを実行することにより、ISPF クライアント・ゲートウェイ接続を検査します。userid は、有効な TSO ユーザー ID に置き換えてください。

```
MODIFY RSED,APPL=IVP ISPF,userid
```

このコマンドは、162 ページの『サービスの検査』で説明した fekfivpi IVP と機能的是に同じです。このコマンドは、次のサンプルのような出力を返します。

F RSED,APPL=IVP ISPF,IBMUSER

```
+FEK900I ISPF IVP: executed on CDFMVS08 -- Tue Sep 13 22:29:28 EDT 2011
+FEK900I ISPF IVP: executed by uid=1(IBMUSER) gid=0(SYS1)
+FEK900I ISPF IVP: using /etc/rdz/rsed.envvars
+FEK900I ISPF IVP: current address space size limit is 2147483647
(2048.0 MB)
+FEK900I ISPF IVP: maximum address space size limit is 2147483647
(2048.0 MB)
+FEK900I ISPF IVP: -----
-----
+FEK900I ISPF IVP: /etc/rdz/ISPF.conf content:
+FEK900I ISPF IVP: -----
-----
+FEK900I ISPF IVP: ispllib=ISP.SISPLOAD
+FEK900I ISPF IVP: ispmllib=ISP.SISPMENU
+FEK900I ISPF IVP: isptlib=ISP.SISPTENU
+FEK900I ISPF IVP: ispplib=ISP.SISPPENU
+FEK900I ISPF IVP: ispslib=ISP.SISPSLIB
+FEK900I ISPF IVP: sysproc=ISP.SISPCLIB,FEK.SFEKPROC
+FEK900I ISPF IVP: -----
-----
+FEK900I ISPF IVP: Host install verification for RSE
+FEK900I ISPF IVP: Review IVP log messages from HOST below :
+FEK900I ISPF IVP: -----
-----
+FEK900I ISPF IVP: Service level 22Feb2011
+FEK900I ISPF IVP: RSE connection and base TSO/ISPF session initializati
on check only
+FEK900I ISPF IVP: *** CHECK : ENVIRONMENT VARIABLES - key variables
displayed below :
+FEK900I ISPF IVP: Server PATH          = ./usr/lpp/java/J6.0/bin:/usr/l
pp/rdz/bin:/usr/lpp/ispf/bin:/bin:/usr/sbin
+FEK900I ISPF IVP: STEPLIB              = NONE
+FEK900I ISPF IVP: Temporary directory = /tmp
+FEK900I ISPF IVP: CGI_ISPHOME         = /usr/lpp/ispf
+FEK900I ISPF IVP: CGI_ISPCONF        = /etc/rdz
+FEK900I ISPF IVP: CGI_ISPWORK        = /var/rdz
+FEK900I ISPF IVP: -----
-----
+FEK900I ISPF IVP: *** CHECK : USS MODULES
+FEK900I ISPF IVP: Checking ISPF Directory : /usr/lpp/ispf
+FEK900I ISPF IVP: Checking modules in /usr/lpp/ispf/bin directory
+FEK900I ISPF IVP: Checking for ISPF configuration file ISPF.conf
+FEK900I ISPF IVP: RC=0
+FEK900I ISPF IVP: MSG: SUCCESSFUL
+FEK900I ISPF IVP: -----
-----
+FEK900I ISPF IVP: *** CHECK : TSO/ISPF INITIALIZATION
+FEK900I ISPF IVP: ( TSO/ISPF session will be initialized )
+FEK900I ISPF IVP: RC=0
+FEK900I ISPF IVP: MSG: SUCCESSFUL
+FEK900I ISPF IVP: -----
-----
+FEK900I ISPF IVP: *** CHECK: Shutting down TSO/ISPF IVP session
+FEK900I ISPF IVP: RC=0
+FEK900I ISPF IVP: MSG: SUCCESSFUL
+FEK900I ISPF IVP: -----
-----
+FEK900I ISPF IVP: Host installation verification completed successfully
+FEK900I ISPF IVP: -----
-----
+FEK901I ISPF IVP Exit code = 0
```

サービスの検査

Developer for System z インストールは、基本およびオプションのサービス用に、いくつかのインストール検査プログラム (IVP) を提供します。IVP スクリプトは、インストール・ディレクトリー (デフォルトでは /usr/lpp/rdz/bin/) に置かれます。

表 20. サービス用の IVP

fekfivpc	167 ページの『(オプション) CARMA 接続』
fekfivpd	165 ページの『RSE デーモン接続』
fekfivpi	166 ページの『ISPF の TSO/ISPF クライアント・ゲートウェイ接続』
fekfivpj	165 ページの『JES ジョブ・モニター接続』
fekfivps	168 ページの『(オプション) SCLMDT 接続』
fekfivpt	164 ページの『TCP/IP のセットアップ』

以下のセクションで説明するタスクでは、ユーザーが z/OS UNIX でアクティブである必要があります。そのためには、**OMVS** TSO コマンド を発行します。TSO に戻するには、**exit** コマンドを使用します。

IVP を実行するユーザー ID には大きな領域サイズが必要です。これは、大量のメモリーを必要とする機能 (Java など) が実行されるからです。領域サイズは、131072 キロバイト (128 メガバイト) 以上に設定してください。

次の例のエラーは、領域サイズが不十分であることを明示していますが、他のエラーが発生する可能性もあります。例えば Java を始動できない場合があります。

```
CEE5213S The signal SIGPIPE was received.
%z/OS UNIX command%: command was killed by signal number 13
  %line-number% **  %REXX command%
+++ RC(137) +++
```

注: IVP テストを開始する前に、Developer for System z 開始タスクをアクティブにしておく必要があります。

IVP の初期化

このセクションのサンプル・コマンドはすべて、特定の環境変数が設定されていることを必要とします。これにより、IVP スクリプトは PATH ステートメントを通じて入手でき、カスタマイズされた構成ファイルのロケーションが判明します。**pwd** コマンドおよび **cd** コマンドを使用して、現行ディレクトリーを検査し、カスタマイズした構成ファイルがあるディレクトリーに変更してください。その後、次のサンプルのように **ivpinit** シェル・スクリプトを使用して RSE 環境変数を設定できます。**\$** は z/OS UNIX プロンプトです。

```
$ pwd
/u/userid
$ cd /etc/rdz
$ ./ivpinit
RSE configuration files located in /etc/rdz --default
added /usr/lpp/rdz/bin to PATH
```

.. /ivpinit の最初のピリオド (.) は、シェルを現行環境で実行するための z/OS UNIX コマンドです。これにより、シェル内で設定された環境変数が、シェルを出た後も有効になります。2 番目のピリオド (.) は現行ディレクトリーを参照しています。

注:

- .. /ivpinit を fekfivp* スクリプトの前に実行しない場合は、呼び出すときにこれらのスクリプトへのパスを指定する必要があります。以下に例を示します。

```
/usr/lpp/rdz/bin/fekfivpr 512 USERID
```

また、.. /ivpinit を最初に実行しない場合、すべての fekfivp* スクリプトは、カスタマイズされた rsed.envvars のロケーションの指定を要求します。

- 一部の IVP テストでは、TCP/IP REXX ソケット API が使用されます。その場合、TCP/IP ロード・ライブラリー (デフォルトは TCPIP.SEZALOAD) が LINKLIST または STEPLIB に指定されている必要があります。これらの IVP テストを実行するには、以下のコマンドが必要になる場合があります。

```
$ EXPORT STEPLIB=$STEPLIB:TCPIP.SEZALOAD
```

既存の STEPLIB に APF 許可のないライブラリーを追加すると、既存の STEPLIB データ・セットの APF 許可が除去されます。

CEE.SCEELKED が LINKLIST または STEPLIB に指定されている場合、TCPIP.SEZALOAD を CEE.SCEELKED の前に配置する必要があります。そうしないと、TCP/IP REXX ソケット呼び出しで OC1 システム異常終了が発生します。

ホスト・システム接続問題の診断方法については、「ホスト構成リファレンス」(SA88-4226) の『構成問題のトラブルシューティング』および Developer for System z Web サイトのサポート・セクション (<http://www-03.ibm.com/software/products/us/en/developerforsystemz/>) の技術情報を参照してください。

ポート可用性

netstat コマンドを発行することにより、JES ジョブ・モニターおよび RSE デーモンのポートが使用可能かどうか検査できます。結果には、それらのサービスで使われているポートが次の例のように表示されます。

IPv4

```
$ netstat
MVS TCP/IP NETSTAT CS VxRy    TCPIP Name: TCPIP      13:57:36
User Id  Conn      Local Socket  Foreign Socket  State
-----
RSED     0000004B  0.0.0.0..4035  0.0.0.0..0      Listen
JMON     00000037  0.0.0.0..6715  0.0.0.0..0      Listen
```

IPv6

```
$ netstat
MVS TCP/IP NETSTAT CS VxRy    TCPIP Name: TCPIP      14:03:35
User Id  Conn      State
-----
RSED     0000004B  Listen
Local Socket:  0.0.0.0..4035
```

```
Foreign Socket: 0.0.0.0..0
JMON      00000037 Listen
Local Socket: 0.0.0.0..6715
Foreign Socket: 0.0.0.0..0
```

TCP/IP のセットアップ

Developer for System z は、TCP/IP が初期化時に正しいホスト名を持っているかどうか依存します。このことは、さまざまな TCP/IP およびリゾルバー構成ファイルが正しくセットアップされていなければならないことを意味しています。TCP/IP およびリゾルバーのセットアップについて詳しくは、「ホスト構成リファレンス」(SA88-4226) の『TCP/IP のセットアップ』を参照してください。次のコマンドを実行することにより、現行の設定値を検査します。

```
fekfivpt
```

注: この IVP では TCPIP **netstat -u** コマンドが発行されますが、ご使用のセキュリティ・ソフトウェアでこのコマンドが実行されないように保護されている場合があります。SERVAUTH クラスの EZB.NETSTAT.mvsname.tcprocname.UP プロファイルを参照してください。

このコマンドは、次の例のような出力を返します。

```
$ fekfivpt

executed on CDFMVS08 -- Wed Jul  2 13:11:54 EDT 2008
executed by uid=1(USERID) gid=0(GROUP)
using /etc/rdz/rsed.envvars

current address space size limit is 1914675200 (1826.0 MB)
maximum address space size limit is 2147483647 (2048.0 MB)

-----
TCP/IP resolver configuration (z/OS UNIX search order):
-----
Resolver Trace Initialization Complete -> 2008/07/02 13:11:54.745964

res_init Resolver values:
Global Tcp/Ip Dataset = None
Default Tcp/Ip Dataset = None
Local Tcp/Ip Dataset  = /etc/resolv.conf
Translation Table     = Default
UserId/JobName         = USERID
Caller API             = LE C Sockets
Caller Mode            = EBCDIC
(L) DataSetPrefix     = TCPIP
(L) HostName          = CDFMVS08
(L) TcpIpJobName       = TCPIP
(L) DomainOrigin      = RALEIGH.IBM.COM
(L) NameServer         = 9.42.206.2
                      9.42.206.3
(L) NsPortAddr        = 53              (L) ResolverTimeout      = 10
(L) ResolveVia        = UDP             (L) ResolverUdpRetries    = 1
(*) Options NDots     = 1
(*) SockNoTestStor    =
(*) AlwaysWto         = NO              (L) MessageCase          = MIXED
(*) LookUp            = DNS LOCAL

res_init Succeeded
res_init Started: 2008/07/02 13:11:54.755363
res_init Ended: 2008/07/02 13:11:54.755371
*****
MVS TCP/IP NETSTAT CS V1R9          TCPIP Name: TCPIP          13:11:54
Tcpip started at 01:28:36 on 06/23/2008 with IPv6 enabled
```



```

-----
host IP address:
-----
hostName=CDFMVS08
hostAddr=9.42.112.75
bindAddr=9.42.112.75
localAddr=9.42.112.75

Success, addresses match

```

RSE デーモン接続

次のコマンドを実行することにより、RSE デーモン接続を検査します。

```
fekfivpd
```

パスワードを求めるプロンプトの後、このコマンドは次の例のような出力を返します。

```

$ fekfivpd

executed on CDFMVS08 -- Wed Jul  2 15:00:27 EDT 2008
executed by uid=1(USERID) gid=0(GROUP)
using /etc/rdz/rsed.envvars

current address space size limit is 1914675200 (1826.0 MB)
maximum address space size limit is 2147483647 (2048.0 MB)

attempting to connect userid USERID using port 4035 ...

Password:
SSL is disabled
connected
8108
570655399
Success

```

SSL に対応した接続をテストする場合、IVP を実行するユーザー ID に、必要なすべての証明書 (Developer for System z 証明書の署名に使用された CA 証明書を含む) へのアクセス権が付与されていることを確認します。この IVP のオペレーター・コマンド・バージョン (F RSED,APPL=IVP DAEMON,userid) は、RSE ホスト・システム用の SSL セットアップを使用するため、エラーが発生しにくくなっています。SSL 関連の一般的なエラーのいくつかを以下にリストします。

- エラー・メッセージ「gsk_environment_init() が失敗しました: 証明書データベースを開くときにエラーが検出されました」を受け取った場合は、IVP を実行するユーザー ID に、必要なすべての証明書へのアクセス権が付与されていることを確認してください。
- エラー・メッセージ「gsk_secure_socket_init() が失敗しました: 証明書検証エラー」を受け取った場合は、署名 CA 証明書が鍵リング上にもあることをチェックしてください。

JES ジョブ・モニター接続

次のコマンドを実行することにより、JES ジョブ・モニター接続を検査します。

```
fekfivpj
```

このコマンドは次のサンプルの内容に似た JES ジョブ・モニター確認応答メッセージを返します (\$ は z/OS UNIX プロンプトです)。

```

$ fekfivpj

executed on CDFMVS08 -- Wed Jul  2 15:00:27 EDT 2008
executed by uid=1(USERID) gid=0(GROUP)
using /etc/rdz/rsed.envvars

current address space size limit is 1914675200 (1826.0 MB)
maximum address space size limit is 2147483647 (2048.0 MB)

testing JES Job Monitor on port 6715...
hostName=CDFMVS08
hostAddr=9.42.112.75
IPv4 is supported
Waiting for JES Job Monitor response...
ACKNOWLEDGE01v03

Success

```

ISPF の TSO/ISPF クライアント・ゲートウェイ接続

次のコマンドを実行することにより、ISPF の TSO/ISPF クライアント・ゲートウェイへの接続を検査します。

```
fekfivpi
```

このコマンドは、ISPF の TSO/ISPF クライアント・ゲートウェイ関連の検査 (変数、HFS モジュール、TSO/ISPF セッションの開始および停止) の結果を返します。出力は、以下の例に示すような内容になるはずです。

```

$ fekfivpi

executed on CDFMVS08 -- Wed Jul  2 15:00:27 EDT 2008
executed by uid=1(USERID) gid=0(GROUP)
using /etc/rdz/rsed.envvars

current address space size limit is 1914675200 (1826.0 MB)
maximum address space size limit is 2147483647 (2048.0 MB)

-----
/etc/rdz/ISPF.conf content:
-----

isplib=ISP.SISPMENU
isptlib=ISP.SISPTENU
ispplib=ISP.SISPPENU
ispslib=ISP.SISPSLIB
ispllib=ISP.SISPLOAD
sysproc=ISP.SISPCLIB,FEK.SFEKPROC

-----
Host install verification for RSE
Review IVP log messages from HOST below :
-----

RSE connection and base TSO/ISPF session initialization check only

*** CHECK : ENVIRONMENT VARIABLES - key variables displayed below :

Server PATH          =
/usr/lpp/java/J6.0/bin:/usr/lpp/rdz/lib:/usr/lpp/ispf/bin:
/bin:/usr/sbin:.

STEPLIB              = FEK.SFEKAUTH:FEK.SFEKLOAD

CGI_ISPHOME          = /usr/lpp/ispf
CGI_ISPCONF          = /etc/rdz

```

```

CGI_ISPWORK = /var/rdz
-----
*** CHECK : USS MODULES
Checking ISPF Directory : /usr/lpp/ispf
Checking modules in /usr/lpp/ispf/bin directory
Checking for ISPF configuration file ISPF.conf
RC=0
MSG: SUCCESSFUL

-----
*** CHECK : TSO/ISPF INITIALIZATION
( TSO/ISPF session will be initialized )
RC=0
MSG: SUCCESSFUL

-----
*** CHECK: Shutting down TSO/ISPF IVP session
RC=0
MSG: SUCCESSFUL

-----
Host installation verification completed successfully
-----

```

注: いずれかの ISPF 検査が失敗した場合は、さらに詳細な情報が表示されます。

fekfivpi には、以下に示すオプションの非定位置パラメーターがあります。

- file** fekfivpi は、数百行に上る大量の出力を生成する場合があります。-file パラメーターは、この出力をファイル \$TMPDIR/fekfivpi.log へ送信します。ここで、\$TMPDIR は rsed.envvars 内の TMPDIR ディレクティブの値で、デフォルトは /tmp です。
- debug** -debug パラメーターは、詳細なテスト出力を作成します。IBM サポートから指示された場合以外は、このオプションを使用しないでください。

(オプション) CARMA 接続

次のコマンドを実行することにより、CARMA への接続を検査します。

```
fekfivpc
```

このコマンドは、次の例に示すような、CARMA 関連の検査の結果を返します。

```

$ fekfivpc

executed on CDFMVS08 -- Fri Aug 20 14:15:46 EDT 2010
executed by uid=1(USERID) gid=0(GROUP)
using /etc/rdz/rsed.envvars

current address space size limit is 140484608 ( 134.0 MB)
maximum address space size limit is 2147483647 (2048.0 MB)

*** /etc/rdz/CRASRV.properties content:
port.start = 5227
port.range = 100
startup.script.name = /usr/lpp/rdz/bin/carma.startup.rex
clist.dsname = *CRASTART
crastart.stub = /usr/lpp/rdz/bin/CRASTART
crastart.configuration.file = /etc/rdz/crastart.endevor.conf
crastart.syslog = Partial
crastart.timeout = 420

*** Creating /tmp/fekfivpc.log

```

```

*** Verifying CARMA installation...
1. Creating CARMA connection (timeout after 60 seconds)
2. Initializing CARMA
3. Retrieving RAM list
   The following RAMs were found
      00 CA Endeavor SCM      Unique ID: COM.IBM.CARMA.ENDEVORRAM
4. Getting customization data for RAM 00
5. Initializing RAM 00
6. Retrieving Repository Instance List
   Found 6 Repository Instance(s)
7. Terminating RAM 00
8. Terminating CARMA

*** IVP Successful!!!!

```

注: IVP が失敗した場合は、/tmp/fekfivpc.log の内容を検査します。このログには RSE と CARMA の間の通信が記録されており、失敗の根本原因の究明に役立つ情報が含まれている可能性があります。

fekfivpc には、以下に示すオプションの非定位置パラメーターがあります。

-noram

デフォルトでは、fekfivpc は、CRADEF VSAM データ・セット内で最初に定義された RAM を始動します。RAM をテストすることが望ましくない場合もあります。例えば、サード・パーティーの RAM が最初にリストされていて、予期しない入力をその RAM が必要とするような場合です。そのような場合、-noram 始動引数を使用して、IVP テストの RAM 固有のステップ (ステップ 4 から 7) を省略することができます。

(オプション) SCLMDT 接続

次のコマンドを実行することにより、SCLM Developer Toolkit への接続を検査します。

```
fekfivps
```

このコマンドは、SCLM Developer Toolkit 関連の検査 (変数、HFS モジュール、REXX ランタイム、TSO/ISPF セッションの開始および停止など) の結果を返し、次の例に示すような出力を表示します。

```
$ fekfivps
```

```

executed on CDFMVS08 -- Wed Jul  2 15:00:27 EDT 2008
executed by uid=1(USERID) gid=0(GROUP)
using /etc/rdz/rsed.envvars

```

```

current address space size limit is 1914675200 (1826.0 MB)
maximum address space size limit is 2147483647 (2048.0 MB)

```

```
-----
/etc/rdz/ISPF.conf content:
-----
```

```

isplib=ISP.SISPMENU
isptlib=ISP.SISPTENU
ispplib=ISP.SISPPENU
ispslib=ISP.SISPSLIB
ispllib=ISP.SISPLOAD
sysproc=ISP.SISPCLIB,FEK.SFEKPROC

```

```

-----
Host install verification for RSE
Review IVP log messages from HOST below :

```

*** CHECK : ENVIRONMENT VARIABLES - key variables displayed below :

Server PATH = /usr/lpp/java/J6.0/bin:/usr/lpp/rdz/lib:/usr/lpp/ispf/bin:
/bin:/usr/sbin:.

STEPLIB = FEK.SFEKAUTH:FEK.SFEKLOAD

CGI_ISPHOME = /usr/lpp/ispf
CGI_ISPCONF = /etc/rdz
CGI_ISPWORK = /var/rdz
_SCLMDT_CONF_HOME = /var/rdz/sclmdt
_SCLMDT_WORK_HOME = /var/rdz
_SCLMDT_TRANTABLE = FEK.#CUST.LSTRANS.FILE

*** CHECK : JAVA PATH SETUP VERIFICATION

RC=0
MSG: SUCCESSFUL

*** CHECK : USS MODULES

Checking ISPF Directory : /usr/lpp/ispf
Checking modules in /usr/lpp/ispf/bin directory
Checking for ISPF configuration file ISPF.conf
Checking install bin Directory : /usr/lpp/rdz/bin
RC=0
MSG: SUCCESSFUL

*** CHECK : REXX RUNTIME ENVIRONMENT

RC=0
MSG: SUCCESSFUL

*** CHECK : TSO/ISPF INITIALIZATION

(TSO/ISPF session will be initialized)
RC=0
MSG: SUCCESSFUL

*** CHECK: Shutting down TSO/ISPF IVP session

RC=0
MSG: SUCCESSFUL

Host installation verification completed successfully

注: いずれかの SCLMDT 検査が失敗した場合は、さらに詳細な情報が表示されます。

fekfivps には、以下に示すオプションの非定位置パラメーターがあります。

-file fekfivps は、数百行に上る大量の出力を生成する場合があります。-file パラメーターは、この出力をファイル \$TMPDIR/fekfivps.log へ送信します。ここで、\$TMPDIR は rsed.envvars 内の TMPDIR ディレクティブの値で、デフォルトは /tmp です。

-debug -debug パラメーターは、詳細なテスト出力を作成します。IBM サポートから指示された場合以外は、このオプションを使用しないでください。

第 9 章 セキュリティー定義

サンプル FEKRACF メンバーをカスタマイズし、実行依頼してください。このメンバーには、Developer for System z 用の基本セキュリティー定義を作成する、サンプルの RACF および z/OS UNIX コマンドが含まれています。

FEKRACF は FEK.#CUST.JCL に置かれます。ただし、FEK.SFEKSAMP(FEKSETUP) ジョブをカスタマイズして実行依頼したときに、別のロケーションを指定した場合は除きます。詳しくは、20 ページの『カスタマイズのセットアップ』を参照してください。

RACF コマンドの詳細については、「RACF コマンド言語解説書」(SA88-8617) を参照してください。

注:

- CA ACF2™ for z/OS を使用しているサイトの場合は、CA サポート・サイト (<https://support.ca.com>) の製品ページを参照し、関連する Developer for System z Knowledge Document TEC492389 を確認してください。この Knowledge Document には、Developer for System z を正しく構成するために必要なセキュリティー・コマンドの詳細が記載されています。
- CA Top Secret® for z/OS を使用しているサイトの場合は、CA サポート・サイト (<https://support.ca.com>) の製品ページを参照し、関連する Developer for System z Knowledge Document TEC492091 を確認してください。この Knowledge Document には、Developer for System z を正しく構成するために必要なセキュリティー・コマンドの詳細が記載されています。

以下のセクションでは、必要なステップ、オプションの構成、および可能な代替策について説明します。

要件およびチェックリスト

セキュリティーのセットアップを完了するために、セキュリティー管理者は、表 21 に示す値を知っている必要があります。これらの値は、前のステップである Developer for System z のインストールとカスタマイズで定義されています。

表 21. セキュリティー・セットアップの変動要素

説明	デフォルト値 正解の入手先	値
Developer for System z 製品の 高位修飾子	FEK SMP/E インストール	
Developer for System z カスタマイズの 高位修飾子	FEK.#CUST FEK.SFEKSAMP(FEKSETUP) (20 ページの『カスタマイズの セットアップ』を参照)	

表 21. セキュリティー・セットアップの変動要素 (続き)

説明	<ul style="list-style-type: none"> デフォルト値 正解の入手先 	値
統合デバッガー開始タスク名	<ul style="list-style-type: none"> DBGMGR FEK.#CUST.PROCLIB (DBGMGR) - 29 ページの『PROCLIB の変更』の説明を参照 	
JES ジョブ・モニター開始タスク名	<ul style="list-style-type: none"> JMON FEK.#CUST.PROCLIB(JMON) (29 ページの『PROCLIB の変更』を参照) 	
RSE デーモン開始タスク名	<ul style="list-style-type: none"> RSED FEK.#CUST.PROCLIB(RSED) (29 ページの『PROCLIB の変更』を参照) 	
アプリケーション ID	<ul style="list-style-type: none"> FEKAPPL /etc/rdz/rsed.envvars (56 ページの『_RSE_JAVAOPTS での追加 Java 始動パラメーターの定義』を参照) 	

以下のリストは、Developer for System z の基本的なセキュリティ・セットアップを完了するために必要な操作の概要を示したものです。以下の各セクションで説明されているように、これらの要件を満たすために、求めるセキュリティ・レベルに応じてさまざまな方式を使用できます。オプションの Developer for System z サービスのセキュリティ・セットアップについては、前述の各セクションを参照してください。

- 173 ページの『セキュリティの設定およびクラスをアクティブにする』
- 174 ページの『Developer for System z ユーザーの OMVS セグメントを定義する』
- 174 ページの『Developer for System z 開始タスクの定義』
- 176 ページの『セキュアな z/OS UNIX サーバーとして RSE を定義する』
- 176 ページの『RSE の MVS プログラム制御ライブラリーを定義する』
- 177 ページの『RSE の PassTicket サポートの定義』
- 179 ページの『RSE のアプリケーション保護の定義』
- 178 ページの『RSE の z/OS UNIX ファイル・アクセス許可の定義』
- 180 ページの『JES コマンド・セキュリティを定義する』
- 182 ページの『統合デバッガーへのアクセスの定義』
- 182 ページの『データ・セット・プロファイルを定義する』
- 186 ページの『セキュリティ設定の検査』

セキュリティの設定およびクラスをアクティブにする

Developer for System z は、セキュアで制御されたホスト・システム環境をクライアントに保証するために、さまざまなセキュリティ・メカニズムを使用しています。そのため、以下のサンプルの RACF コマンドで示すように、いくつかのクラスおよびセキュリティ設定をアクティブにする必要があります。

- 現行の設定を表示する
 - SETROPTS LIST
- z/OS UNIX、デジタル証明書プロファイル、および統合デバッガーのファシリティ・クラスをアクティブにする
 - SETROPTS GENERIC(FACILITY)
 - SETROPTS CLASSACT(FACILITY) RACLIST(FACILITY)
- 開始タスク定義をアクティブにする
 - SETROPTS GENERIC(STARTED)
 - RDEFINE STARTED ** STDATA(USER(=MEMBER) GROUP(STCGROUP) TRACE(YES))
 - SETROPTS CLASSACT(STARTED) RACLIST(STARTED)
- JES ジョブ・モニターのコンソール・セキュリティをアクティブにする
 - SETROPTS GENERIC(CONSOLE)
 - SETROPTS CLASSACT(CONSOLE) RACLIST(CONSOLE)
- JES ジョブ・モニターのオペレーター・コマンド保護をアクティブにする
 - SETROPTS GENERIC(OPERCMDS)
 - SETROPTS CLASSACT(OPERCMDS) RACLIST(OPERCMDS)
- RSE の z/OS UNIX ファイル・アクセス許可をアクティブにする
 - o SETROPTS GENERIC(UNIXPRIV)
 - o SETROPTS CLASSACT(UNIXPRIV) RACLIST(UNIXPRIV)
- RSE のアプリケーション保護をアクティブにする
 - SETROPTS GENERIC(APPL)
 - SETROPTS CLASSACT(APPL) RACLIST(APPL)
- RSE の PassTicket を使用したセキュアなサインオンをアクティブにする
 - SETROPTS GENERIC(PTKTDATA)
 - SETROPTS CLASSACT(PTKTDATA) RACLIST(PTKTDATA)
- 信頼されたコードだけを RSE がロードできるように、プログラム制御をアクティブにする
 - RDEFINE PROGRAM ** ADDMEM('SYS1.COMDLIB'//NOPADCHK) UACC(READ)
 - SETROPTS WHEN(PROGRAM)

注: すでに PROGRAM クラス内に * プロファイルがある場合は、** プロファイルを作成しないでください。セキュリティ・ソフトウェアによって使用される検索パスが、分かりにくく、複雑なものになります。その場合は、既存の * 定義と新しい ** 定義をマージする必要があります。 ** プロファイルの使用については、「*Security Server RACF セキュリティ管理者のガイド*」(SA88-8613) に説明があります。

重要: 「WHEN PROGRAM」がアクティブの場合、一部の製品 (FTP など) はプログラムで制御することが必要です。このプログラム制御は、実動システム上でアクティブにする前にテストしてください。

- (オプション) X.509 HostIdMappings および拡張 Port Of Entry (POE) サポートをアクティブにする
 - SETROPTS GENERIC(SERVAUTH)
 - SETROPTS CLASSACT(SERVAUTH) RACLIST(SERVAUTH)

Developer for System z ユーザーの OMVS セグメントを定義する

Developer for System z のユーザーごとに、有効なゼロ以外の z/OS UNIX ユーザー ID (UID)、ホーム・ディレクトリー、およびシェル・コマンドを指定する RACF OMVS セグメントまたは同等のものを定義する必要があります。また、ユーザーのデフォルト・グループも、グループ ID を持つ OMVS セグメントを必要とします。

オプションの統合デバッガーを使用するときには、デバッグするアプリケーションをアクティブ化したユーザー ID、およびそのデフォルト・グループにも、有効な RACF OMVS セグメントまたは同等のものがが必要です。

以下のサンプルの RACF コマンドでは、#userid、#user-identifier、#group-name、および #group-identifier の各プレースホルダーを実際の値に置き換えてください。

- ALTUSER #userid
OMVS(UID(#user-identifier) HOME(/u/#userid) PROGRAM(/bin/sh) NOASSIZEMAX)
- ALTGROUP #group-name OMVS(GID(#group-identifier))

Developer for System z 開始タスクの定義

以下のサンプル RACF コマンドは、保護されたユーザー ID (STCDBM、STCJMON、STCRSE) およびそれらに割り当てられた FEKD、DBGMGR、JMON、および RSED 開始タスクを作成します。 #group-id および #user-id-* プレースホルダーは、有効な OMVS ID に置き換えてください。

- ADDGROUP STCGROUP OMVS(AUTOGID)
DATA('GROUP WITH OMVS SEGMENT FOR STARTED TASKS')
- ADDUSER STCDBM DFLTGRP(STCGROUP) NOPASSWORD NAME('RDZ - DEBUG MANAGER')
OMVS(AUTOUID HOME(/tmp) PROGRAM(/bin/sh))
DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
- ADDUSER STCJMON DFLTGRP(STCGROUP) NOPASSWORD NAME('RDZ - JES JOBMONITOR')
OMVS(AUTOUID HOME(/tmp) PROGRAM(/bin/sh))
DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
- ADDUSER STCRSE DFLTGRP(STCGROUP) NOPASSWORD NAME('RDZ - RSE DAEMON')
OMVS(AUTOUID HOME(/tmp) PROGRAM(/bin/sh) ASSIZEMAX(2147483647))
DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
- RDEFINE STARTED DBGMGR.* DATA('RDZ - DEBUG MANAGER')
STDATA(USER(STCDBM) GROUP(STCGROUP) TRUSTED(NO))
- RDEFINE STARTED JMON.* DATA('RDZ - JES JOBMONITOR')
STDATA(USER(STCJMON) GROUP(STCGROUP) TRUSTED(NO))
- RDEFINE STARTED RSED.* DATA('RDZ - RSE DAEMON')
STDATA(USER(STCRSE) GROUP(STCGROUP) TRUSTED(NO))

- SETROPTS RACLIST(STARTED) REFRESH

注:

- NOPASSWORD キーワードを指定することにより、開始タスクのユーザー ID が必ず保護されるようにしてください。
- RSE サーバーが固有の OMVS uid を持つようにしてください (その uid へ付与される z/OS UNIX 関連の特権のため)。
- RSE デーモンを適切に動作させるためには、大きなサイズのアドレス・スペース (2 GB) が必要です。この値は、ユーザー ID STCRSE の OMVS セグメントの ASSIZEMAX 変数で設定してください。この値を設定することにより、SYS1.PARMLIB(BPXPRMxx) で MAXASSIZE が変更されても、RSE デーモンには必要な領域サイズが確実に設定されます。
- RSE を適切に動作させるためには、多数のスレッドも必要です。スレッド数の限度は、ユーザー ID STCRSE の OMVS セグメントの THREADSMAX 変数で設定できます。この限度を設定することにより、SYS1.PARMLIB(BPXPRMxx) で MAXTHREADS または MAXTHREADTASKS が変更されても、RSE には必要なスレッド数の限度が確実に設定されます。スレッド数の限度として適切な値を決定するには、「ホスト構成リファレンス」(SA88-4226) の『チューニングに関する考慮事項』を参照してください。
- ユーザー ID STCJMON も、OMVS セグメントで THREADSMAX を設定する対象となります。これは、JES ジョブ・モニターがクライアント接続ごとにスレッドを使用するためです。
- 統合デバッガー開始タスク (DBGMR) は、オプションの統合デバッガー機能によってのみ使用されます。

STCRSE ユーザー IDを制限することを検討してください。RESTRICTED 属性を持つユーザーは、保護された (MVS) リソースには、特にアクセスを許可されている場合以外、アクセスできません。

ALTUSER STCRSE RESTRICTED

制限されたユーザーが「その他の」許可ビットによって z/OS UNIX ファイル・システム・リソースにアクセスできないよう、UNIXPRIV クラス内に RESTRICTED.FILESYS.ACCESS プロファイルを UACC(NONE) で定義します。ユーザー ID の制限の詳細については、「*Security Server RACF セキュリティー管理者のガイド*」(SA88-8613) を参照してください。

重要: 制限されたユーザー ID を使用する場合は、TSO の **PERMIT** コマンドか z/OS UNIX の **setfac1** コマンドを使用して、リソースにアクセスする許可を明示的に追加してください。そのようなリソースには、Developer for System z 資料で UACC が使用されているリソース (PROGRAM クラス内の ** プロファイルなど) や、共通の z/OS UNIX 規則 (全員が Java ライブラリーの読み取り権限と実行権限を持つなど) に依存するリソースなどがあります。アクセス権限は、実動システムでアクティブにする前にテストしてください。

セキュアな z/OS UNIX サーバーとして RSE を定義する

RSE は、クライアントのスレッドのセキュリティ環境を作成または削除するためには、BPX.SERVER プロファイルに対する UPDATE アクセス権を必要とします。このプロファイルが定義されていない場合は、UID(0) が RSE に必要です。このステップは、クライアントが接続可能になるために必要です。

統合デバッガーは、デバッグ・スレッドのセキュリティ環境を作成または削除するために、BPX.SERVER プロファイルに対する UPDATE アクセス権を必要とします。このプロファイルが定義されない場合、STCDBM 開始タスク・ユーザー ID 用に UID(0) が必要です。この許可は、オプションの統合デバッガー機能を使用する場合にのみ必要となります。

- RDEFINE FACILITY BPX.SERVER UACC(NONE)
- PERMIT BPX.SERVER CLASS(FACILITY) ACCESS(UPDATE) ID(STCRSE)
- PERMIT BPX.SERVER CLASS(FACILITY) ACCESS(UPDATE) ID(STCDBM)
- SETROPTS RACLIST(FACILITY) REFRESH

重要: BPX.SERVER プロファイルを定義すると、z/OS UNIX 全体が UNIX レベルのセキュリティから、より安全な z/OS UNIX レベルのセキュリティに切り替わります。この切り替えによって、他の z/OS UNIX アプリケーションと操作が影響を受ける場合もあります。セキュリティは、実動システム上でアクティブにする前にテストしてください。各種セキュリティ・レベルの詳細については、「UNIX System Services 計画」(GA88-8639)を参照してください。

RSE の MVS プログラム制御ライブラリーを定義する

BPX.SERVER に対する権限を持つサーバーは、クリーンなプログラム制御環境で実行する必要があります。この要件は、RSE によって呼び出されるすべてのプログラムも、プログラムで制御する必要があることを意味します。MVS ロード・ライブラリーの場合、プログラム制御はセキュリティ・ソフトウェアによって管理されます。このステップは、クライアントが接続可能になるために必要です。

RSE は、システム (SYS1.LINKLIB)、言語環境プログラムのランタイム (CEE.SCEERUN*)、および ISPF の TSO/ISPF クライアント・ゲートウェイ (ISP.SISPLOAD) ロード・ライブラリーを使用します。

- RALTER PROGRAM ** UACC(READ) ADDMEM('SYS1.LINKLIB'//NOPADCHK)
- RALTER PROGRAM ** UACC(READ) ADDMEM('SYS1.CSSLIB'//NOPADCHK)
- RALTER PROGRAM ** UACC(READ) ADDMEM('CEE.SCEERUN'//NOPADCHK)
- RALTER PROGRAM ** UACC(READ) ADDMEM('CEE.SCEERUN2'//NOPADCHK)
- RALTER PROGRAM ** UACC(READ) ADDMEM('ISP.SISPLOAD'//NOPADCHK)
- SETROPTS WHEN(PROGRAM) REFRESH

注: すでに PROGRAM クラス内に * プロファイルがある場合は、** プロファイルを使用しないでください。このプロファイルを使用すると、セキュリティ・ソフトウェアで使用する検索パスが分かりにくく複雑になります。その場合は、既存の

* 定義と新しい ** 定義をマージする必要があります。 ** プロファイルの使用については、「*Security Server RACF セキュリティー管理者のガイド*」(SA88-8613) に説明があります。

オプションのサービスを使用できるようにするには、以下の前提条件の追加ライブラリーがプログラムで制御されるようにする必要があります。このリストには、Developer for System z と対話する製品 (IBM File Manager など) に固有のデータ・セットは含まれていません。

- 代替 REXX ランタイム・ライブラリー、SCLM Developer Toolkit 用
 - REXX.*.SEAGALT
- システム・ロード・ライブラリー、SSL 暗号化用
 - SYS1.SIEALNKE
- Developer for System z ライブラリー、統合デバッガー用
 - FEK.SFEKAUTH

注: LPA 配置用に設計されたライブラリーは、LINKLIST または STEPLIB によってアクセスされる場合、追加のプログラム制御権限も必要とします。この資料では、以下の LPA ライブラリーの使用法について説明します。

- ISPF、TSO/ISPF クライアント・ゲートウェイ用
 - ISP.SISPLPA
- REXX ランタイム・ライブラリー、SCLM Developer Toolkit 用
 - REXX.*.SEAGLPA
- Developer for System z、CARMA 用
 - FEK.SFEKLPA

RSE の PassTicket サポートの定義

クライアントのパスワードまたは X.509 証明書などのその他の識別手段は、接続時に ID を検査するためにのみ使用されます。その後は、スレッド・セキュリティーを維持するために PassTicket が使用されます。このステップは、クライアントが接続可能になるために必要です。

PassTicket は、有効期間が約 10 分のシステム生成パスワードです。パスチケットは、秘密鍵に基づいて生成されます。この鍵は 64 ビットの数値 (16 個の 16 進文字) です。以下のサンプルの RACF コマンドでは、key16 プレースホルダーをユーザー指定の 16 文字の 16 進数ストリング (0 から 9 と A から F までの文字から成る) に置き換えてください。

- RDEFINE PTKTDATA FEKAPPL UACC(NONE) SSIGNON(KEYMASKED(key16))
APPLDATA('NO REPLAY PROTECTION - DO NOT CHANGE')
DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
- RDEFINE PTKTDATA IRRPTAUTH.FEKAPPL.* UACC(NONE)
DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
- PERMIT IRRPTAUTH.FEKAPPL.* CLASS(PTKTDATA) ACCESS(UPDATE) ID(STCRSE)
- SETROPTS RACLIST(PTKTDATA) REFRESH

RSE は、FEKAPPL 以外のアプリケーション ID の使用をサポートしています。これをアクティブにするには、56 ページの『_RSE_JAVAOPTS での追加 Java 始動パラメーターの定義』の説明に従って、rsed.envvars 内の「APPLID=FEKAPPL」オプション

ンのコメントを外してカスタマイズします。PTKTDATA クラス定義は、RSE が使用する実際のアプリケーション ID と一致している必要があります。

OMVSAPPL はアプリケーション ID として使用しないでください。これは、大部分の z/OS UNIX アプリケーションの秘密鍵を公開するからです。また、デフォルトの MVS アプリケーション ID (MVS の直後にシステムの SMF ID を続けたもの) も使用しないでください。これは、ユーザー・バッチ・ジョブを含む大部分の MVS アプリケーションの秘密鍵を公開するからです。

注:

- PTKTDATA クラスが既に定義されている場合は、上記のリストにあるプロファイルを作成する前に、それが総称クラスとして定義されていることを確認してください。PTKTDATA クラス内の総称文字のサポートは、PassTicket に Java インターフェースが導入された z/OS リリース 1.7 からの新機能です。
- RSE が PassTicket を生成できるユーザー ID を制限するには、IRRPTAUTH.FEKAPPL.* 定義の中のワイルドカード (*) を、有効なユーザー ID マスクで置き換えます。
- RACF の設定によっては、プロファイルを定義しているユーザーが、そのプロファイルのアクセス・リストにも入っている場合があります。PTKTDATA プロファイルのこの許可を削除してください。
- RSE が提示した PassTicket を JES ジョブ・モニターが評価できるようにするには、JES ジョブ・モニターと RSE が、同じアプリケーション ID を持っている必要があります。JES ジョブ・モニターでは、アプリケーション ID は FEJJC�FG 構成ファイルに APPLID ディレクティブと一緒に設定されます。
- システムに暗号製品がインストールされており、使用可能になっている場合、保護されたサインオン・アプリケーション鍵を暗号化して、保護を強化することができます。そのためには、KEYMASKED ではなく KEYENCRYPTED キーワードを使用します。詳細については、「*Security Server RACF セキュリティー管理者のガイド*」(SA88-8613) を参照してください。

重要: PassTicket が正しくセットアップされていないと、クライアントの接続要求は失敗します。

RSE の z/OS UNIX ファイル・アクセス許可の定義

MODIFY LOGS オペレーター・コマンドは、RSED 開始タスクのユーザー ID を使用して、ホスト・ログおよびセットアップ情報を収集します。また、デフォルトで、セキュア・ファイル・アクセス許可 (所有者のみがアクセス権を保持する) を使用して、ユーザー・ログ・ファイルが作成されます。セキュア・ユーザー・ログ・ファイルを収集できるようにするには、RSED 開始タスクのユーザー ID が、それらのファイルを読み取る許可を持っていない限りなりません。

MODIFY LOGS オペレーター・コマンドの OWNER 引数を指定すると、指定されたユーザー ID が、収集されたデータの所有者になります。所有権を変更するには、RSED 開始タスクのユーザー ID が、CHOWN z/OS UNIX サービスを使用する許可を持っていない限りなりません。

- RDEFINE UNIXPRIV SUPERUSER.FILESYS UACC(NONE) DATA('OVERRIDE UNIX FILE ACCESS RESTRICTIONS')
- RDEFINE UNIXPRIV SUPERUSER.FILESYS.CHOWN UACC(NONE) DATA('OVERRIDE UNIX CHANGE OWNER RESTRICTIONS')
- PERMIT SUPERUSER.FILESYS CLASS(UNIXPRIV) ACCESS(READ) ID(STCRSE)
- PERMIT SUPERUSER.FILESYS.CHOWN CLASS(UNIXPRIV) ACCESS(READ) ID(STCRSE)
- SETROPTS RACLIST(UNIXPRIV) REFRESH

SUPERUSER.FILESYS.ACLOVERRIDE プロファイルが定義されている場合、ACL (アクセス制御リスト) で定義されているアクセス許可は、SUPERUSER.FILESYS を介して付与された許可よりも優先される点に注意してください。ACL 定義をバイパスする場合、RSED 開始タスクのユーザー ID には、SUPERUSER.FILESYS.ACLOVERRIDE プロファイルに対する READ アクセス権が必要になります。

RSE のアプリケーション保護の定義

クライアントがログオンするときに、RSE デーモンはユーザーがアプリケーションの使用を許可されていることを検証します。

- RDEFINE APPL FEKAPPL UACC(READ) DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
- SETROPTS RACLIST(APPL) REFRESH

注:

- 177 ページの『RSE の PassTicket サポートの定義』で詳しく説明するように、RSE は FEKAPPL 以外のアプリケーション ID の使用をサポートしています。APPL クラス定義は、RSE が使用する実際のアプリケーション ID と一致している必要があります。
- アプリケーション ID が APPL クラスに定義されていない場合、クライアント接続要求は成功します。
- アプリケーション ID が定義されていて、ユーザーにプロファイルへの READ アクセス権がない場合にのみ、クライアント接続要求は失敗します。

RSE の z/OS UNIX プログラム制御ファイルを定義する

BPX.SERVER に対する権限を持つサーバーは、クリーンなプログラム制御環境で実行する必要があります。この要件は、RSE によって呼び出されるすべてのプログラムも、プログラムで制御する必要があることを意味します。z/OS UNIX ファイルの場合、プログラム制御は **extattr** コマンドによって管理されます。このコマンドを実行するには、FACILITY クラス内の BPX.FILEATTR.PROGCTL に対する READ アクセス権を持つか、または UID(0) であることが必要です。

RSE サーバーは、RACF の Java 共用ライブラリー (/usr/lib/libIRRRacf*.so) を使用します。

- extattr +p /usr/lib/libIRRRacf*.so

注:

- z/OS 1.9 以降、/usr/lib/libIRRRacf*.so は SMP/E による RACF のインストール時にプログラム制御モードでインストールされます。

- z/OS 1.10 以降では、/usr/lib/libIRRRacf*.so は、基本の z/OS に付属している SAF に含まれているため、RACF を持っていないお客様も使用できます。
- RACF 以外のセキュリティ製品を使用している場合は、セットアップが異なることがあります。詳細については、ご使用のセキュリティ製品の資料を参照してください。
- Developer for System z の SMP/E インストールは、内部 RSE プログラムのプログラム制御ビットを設定します。
- プログラム制御ビットの現在の状況を表示するには、z/OS UNIX コマンド **ls -Eog** を使用します。2 番目のストリング内に英字の **p** が表示される場合、そのファイルはプログラムで制御されています。

```
$ ls -Eog /usr/lib/libIRRRacf*.so
-rwxr-xr-x  aps-  2      69632 Oct  5  2007 /usr/lib/libIRRRacf.so
-rwxr-xr-x  aps-  2      69632 Oct  5  2007 /usr/lib/libIRRRacf64.so
```

JES コマンド・セキュリティを定義する

JES ジョブ・モニターは、ユーザーが要求したすべての JES オペレーター・コマンドを、拡張 MCS (EMCS) コンソールを通じて発行します。このコンソールの名前は、38 ページの『FEJCNFG、JES ジョブ・モニター構成ファイル』にあるように、`CONSOLE_NAME` ディレクティブによって制御されます。

以下のサンプルの RACF コマンドは、Developer for System z ユーザーに、限定された一式の JES コマンド (保留、保留解除、キャンセル、およびページ) に対する条件付きアクセス権を与えます。ユーザーは、JES ジョブ・モニターによってコマンドを発行した場合にのみ、実行権限を持ちます。`#console` プレースホルダーは、実際のコンソール名に置き換えてください。

- `RDEFINE OPERCMDS MVS.MCSOPER.#console UACC(READ) DATA('RATIONAL DEVELOPER FOR SYSTEM Z')`
- `RDEFINE OPERCMDS JES%.** UACC(NONE)`
- `PERMIT JES%.** CLASS(OPERCMDS) ACCESS(UPDATE) WHEN(CONSOLE(JMON)) ID(*)`
- `SETROPTS RACLIST(OPERCMDS) REFRESH`

注:

- コンソールの使用は、`MVS.MCSOPER.#console` プロファイルが定義されていない場合に許可されます。
- `WHEN(CONSOLE(JMON))` が機能するためには、`CONSOLE` クラスがアクティブでなければなりません。また、`CONSOLE` クラス内に `EMCS` コンソールがあるかどうかについての実際のプロファイル検査はありません。
- `WHEN(CONSOLE(JMON))` 文節内で、`JMON` を実際のコンソール名に置き換えないでください。`JMON` キーワードは、コンソール名ではなく、入り口点アプリケーションを表しています。

重要: ご使用のセキュリティ・ソフトウェアで汎用アクセス `NONE` を使用して JES コマンドを定義すると、他のアプリケーションや操作に影響が出る場合があります。セキュリティは、実動システム上でアクティブにする前にテストしてください。

表 22 および表 23 は、JES2 および JES3 について発行されたオペレーター・コマンドと、それらを保護するために使用できる個別セキュリティー・プロファイルを示しています。

表 22. JES2 ジョブ・モニターのオペレーター・コマンド

アクション	コマンド	OPERCMDS プロファイル	必要なアクセス権
保留	\$Hx(jobid) x = {J、S、または T}	jesname.MODIFYHOLD.BAT jesname.MODIFYHOLD.STC jesname.MODIFYHOLD.TSU	UPDATE
保留解除	\$Ax(jobid) x = {J、S、または T}	jesname.MODIFYRELEASE.BAT jesname.MODIFYRELEASE.STC jesname.MODIFYRELEASE.TSU	UPDATE
キャンセル	\$Cx(jobid) x = {J、S、または T}	jesname.CANCEL.BAT jesname.CANCEL.STC jesname.CANCEL.TSU	UPDATE
パージ	\$Cx(jobid),P x = {J、S、または T}	jesname.CANCEL.BAT jesname.CANCEL.STC jesname.CANCEL.TSU	UPDATE

表 23. JES3 ジョブ・モニターのオペレーター・コマンド

アクション	コマンド	OPERCMDS プロファイル	必要なアクセス権
保留	*F,J=jobid,H	jesname.MODIFY.JOB	UPDATE
保留解除	*F,J=jobid,R	jesname.MODIFY.JOB	UPDATE
キャンセル	*F,J=jobid,C	jesname.MODIFY.JOB	UPDATE
パージ	*F,J=jobid,C	jesname.MODIFY.JOB	UPDATE

注:

- 「保留」、「保留解除」、「キャンセル」、「パージ」の各 JES オペレーター・コマンドと「JCL の表示」コマンドは、クライアント・ユーザー ID が所有しているスプール・ファイルに対してのみ実行できます。ただし、JES ジョブ・モニター構成ファイル内で、LIMIT_COMMANDS= が値 LIMITED または NOLIMIT に指定されている場合は除きます。詳しくは、「ホスト構成リファレンス」(SA88-4226) の『ジョブに対するアクション - ターゲットの制限』を参照してください。
- ユーザーは、JES ジョブ・モニター構成ファイル内で LIMIT_VIEW=USERID が定義されている場合を除き、すべてのスプール・ファイルを参照できます。詳しくは、「ホスト構成リファレンス」(SA88-4226) の『スプール・ファイルへのアクセス』を参照してください。
- これらのオペレーター・コマンドに関する許可がない場合でも、ユーザーは、これらのリソースを保護できるプロファイル (JESINPUT, JESJOBS クラスおよび JESSPOOL クラス内のプロファイルなど) に対して十分な権限を持っていれば、JES ジョブ・モニターを介してジョブを実行依頼し、ジョブ出力を読み取ることができます。

TSO セッションから JMON コンソールを作成することによって JES ジョブ・モニター・サーバーの ID を装うことは、セキュリティー・ソフトウェアによって防止されます。コンソールを作成できる場合でも、例えば JES ジョブ・モニターと TSO とではエントリー・ポイントが異なります。この資料で説明されているとおり、セキュリティーがセットアップされており、ユーザーが他の手段によって JES コマンドに対する権限を持っていない場合は、そのコンソールから発行された JES コマンドはセキュリティー検査で不合格になります。

統合デバッガーへのアクセスの定義

ユーザーが問題プログラム状態のプログラムのデバッグに統合デバッガーを使用できるようにするには、リストされている AQE.AUTHDEBUG.* プロファイルのいずれかに対する READ アクセス権が必要です。AQE.AUTHDEBUG.AUTHPGM プロファイルへのアクセスが許可されているユーザーは、APF 許可済みのプログラムをデバッグすることもできます。#apf プレースホルダーは、許可済みプログラムのデバッグが許可されているユーザーの、有効なユーザー ID または RACF グループ名に置き換えてください。

- RDEFINE FACILITY AQE.AUTHDEBUG.STDPGM UACC(NONE)
- PERMIT AQE.AUTHDEBUG.STDPGM CLASS(FACILITY) ACCESS(READ) ID(*)
- RDEFINE FACILITY AQE.AUTHDEBUG.AUTHPGM UACC(NONE)
- PERMIT AQE.AUTHDEBUG.AUTHPGM CLASS(FACILITY) ACCESS(READ) ID(#apf)
- SETROPTS RACLIST(FACILITY) REFRESH

注: バージョン 9.1.1 より前のバージョンの Developer for System z では、現在使用されていない別の FACILITY クラス・プロファイル AQE.AUTHDEBUG.WRITEBUFFER を使用していました。これは、ご使用のホスト・システムにバージョン 9.1.1 以上の Developer for System z しかない場合には、削除できます。

データ・セット・プロファイルを定義する

ほとんどの Developer for System z データ・セットでは、ユーザーの場合は READ アクセス権、システム・プログラマーの場合は ALTER で十分です。#sysprog プレースホルダーは、有効なユーザー ID または RACF グループ名に置き換えてください。また、正しいデータ・セット名については、製品をインストールおよび構成したシステム・プログラマーに問い合わせてください。FEK は、インストール時に使用されたデフォルトの高位修飾子で、FEK.#CUST はカスタマイズ・プロセスで作成されたデータ・セットのデフォルトの高位修飾子です。

- ADDGROUP (FEK) OWNER(IBMUSER) SUPGROUP(SYS1)
DATA('RATIONAL DEVELOPER FOR SYSTEM Z - HLQ STUB')
- ADDSD 'FEK.*.*' UACC(READ)
DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
- PERMIT 'FEK.*.*' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
- SETROPTS GENERIC(DATASET) REFRESH

注:

- FEK.SFEKAUTH には APF 許可があるため、このデータ・セットを更新されないよう保護してください。FEK.SFEKLOAD および FEK.SFEKLPA はプログラム制御されているため、これらも保護することをお勧めします。

- この資料内および FEKRACF ジョブ内のサンプル・コマンドでは、拡張総称命名 (EGN) がアクティブであることを想定しています。EGN がアクティブな場合、** 修飾子を使用して、DATASET クラス内の任意の数の修飾子を表すことができます。使用しているシステムで EGN がアクティブでない場合は、** を * に置き換えてください。EGN の詳細については、「*Security Server RACF セキュリティ管理者のガイド*」(SA88-8613) を参照してください。

一部のオプションの Developer for System z コンポーネントには、追加のセキュリティ・データ・セット・プロファイルが必要です。#sysprog、#ram-developer、および #cicsadmin の各プレースホルダーは、有効なユーザー ID または RACF グループ名に置き換えてください。

- SCLM Developer Toolkit のロング/ショート・ネーム変換を使用している場合は、ユーザーにマッピング VSAM の FEK.#CUST.LSTRANS.FILE に対する UPDATE アクセス権が必要です。
 - ADDSD 'FEK.#CUST.LSTRANS.*.*' UACC(UPDATE)
DATA('RATIONAL DEVELOPER FOR SYSTEM Z - SCLMDT')
 - PERMIT 'FEK.#CUST.LSTRANS.*.*' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
 - SETROPTS GENERIC(DATASET) REFRESH
- CARMA RAM (Repository Access Manager) 開発者には、CARMA VSAM である FEK.#CUST.CRA* に対する UPDATE アクセス権が必要です。
 - ADDSD 'FEK.#CUST.CRA*.*' UACC(READ)
DATA('RATIONAL DEVELOPER FOR SYSTEM Z - CARMA')
 - PERMIT 'FEK.#CUST.CRA*.*' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
 - PERMIT 'FEK.#CUST.CRA*.*' CLASS(DATASET) ACCESS(UPDATE) ID(#ram-developer)
 - SETROPTS GENERIC(DATASET) REFRESH
- Application Deployment Manager の CICS リソース定義 (CRD) サーバーを使用している場合は、CICS 管理者に、CRD リポジトリ VSAM に対する UPDATE アクセス権が必要です。
 - ADDSD 'FEK.#CUST.ADNREP*.*' UACC(READ)
DATA('RATIONAL DEVELOPER FOR SYSTEM Z - ADN')
 - PERMIT 'FEK.#CUST.ADNREP*.*' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
 - PERMIT 'FEK.#CUST.ADNREP*.*' CLASS(DATASET) ACCESS(UPDATE) ID(#cicsadmin)
 - SETROPTS GENERIC(DATASET) REFRESH
- Application Deployment Manager のマニフェスト・リポジトリが定義されている場合は、すべての CICS Transaction Server ユーザーに、マニフェスト・リポジトリ VSAM に対する UPDATE アクセス権が必要です。
 - ADDSD 'FEK.#CUST.ADNMAN*.*' UACC(UPDATE)
DATA('RATIONAL DEVELOPER FOR SYSTEM Z - ADN')
 - PERMIT 'FEK.#CUST.ADNMAN*.*' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
 - SETROPTS GENERIC(DATASET) REFRESH

READ アクセス権も制御対象とする、さらにセキュアなセットアップのためには、以下のサンプル RACF コマンドを使用します。

- uacc(none) データ・セット保護
 - ADDGROUP (FEK)
DATA('RATIONAL DEVELOPER FOR SYSTEM Z - HLQ STUB')
OWNER(IBMUSER) SUPGROUP(SYS1)"
 - ADDSD 'FEK.*.*' UACC(NONE)
DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
 - ADDSD 'FEK.SFEKAUTH' UACC(NONE)
DATA('RATIONAL DEVELOPER FOR SYSTEM Z')

```

- ADDSD 'FEK.SFEKLOAD' UACC(NONE)
  DATA('RATIONAL DEVELOPER FOR SYSTEM Z')

  ADDSD 'FEK.SFEKLMOD' UACC(NONE)
  DATA('RATIONAL DEVELOPER FOR SYSTEM Z')

-
  ADDSD 'FEK.SFEKPROC' UACC(NONE)
  DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
- ADDSD 'FEK.#CUST.PARMLIB' UACC(NONE)
  DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
- ADDSD 'FEK.#CUST.CNTL' UACC(NONE)
  DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
-
  ADDSD 'FEK.#CUST.SQL' UACC(NONE)
  DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
- ADDSD 'FEK.#CUST.LSTRANS.*.*' UACC(NONE)
  DATA('RATIONAL DEVELOPER FOR SYSTEM Z - SCLMDT')
- ADDSD 'FEK.#CUST.CRA*.*' UACC(NONE)
  DATA('RATIONAL DEVELOPER FOR SYSTEM Z - CARMA')
- ADDSD 'FEK.#CUST.ADNREP*.*' UACC(NONE)
  DATA('RATIONAL DEVELOPER FOR SYSTEM Z - ADN')
- ADDSD 'FEK.#CUST.ADNMAN*.*' UACC(NONE)
  DATA('RATIONAL DEVELOPER FOR SYSTEM Z - ADN')
• システム・プログラマーにすべてのライブラリーの管理を許可する
- PERMIT 'FEK.*.*' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
- PERMIT 'FEK.SFEKAUTH' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
- PERMIT 'FEK.SFEKLOAD' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
- PERMIT 'FEK.SFEKLMOD' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
- PERMIT 'FEK.SFEKPROC' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
- PERMIT 'FEK.#CUST.PARMLIB' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
- PERMIT 'FEK.#CUST.CNTL' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
-
  PERMIT 'FEK.#CUST.SQL' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
- PERMIT 'FEK.#CUST.LSTRANS.*.*' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
- PERMIT 'FEK.#CUST.CRA*.*' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
- PERMIT 'FEK.#CUST.ADNREP*.*' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
- PERMIT 'FEK.#CUST.ADNMAN*.*' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
• クライアントにロードおよび exec ライブラリーへのアクセスを許可する
- PERMIT 'FEK.SFEKAUTH' CLASS(DATASET) ACCESS(READ) ID(*)
- PERMIT 'FEK.SFEKLOAD' CLASS(DATASET) ACCESS(READ) ID(*)
- PERMIT 'FEK.SFEKPROC' CLASS(DATASET) ACCESS(READ) ID(*)
- PERMIT 'FEK.#CUST.CNTL' CLASS(DATASET) ACCESS(READ) ID(*)
-
  PERMIT 'FEK.#CUST.SQL' CLASS(DATASET) ACCESS(READ) ID(*)

```

注: FEK.SFEKLPA に許可は必要ありません。LPA 内に存在するすべてのコードには、全員がアクセスできるからです。

- 統合デバッガーに、ロード・ライブラリーへのアクセスを許可します。
 - PERMIT 'FEK.SFEKAUTH' CLASS(DATASET) ACCESS(READ) ID(STCDBM)
- JES ジョブ・モニターにロードおよびパラメーター・ライブラリーへのアクセスを許可する

- PERMIT 'FEK.SFEKAUTH' CLASS(DATASET) ACCESS(READ) ID(STCJMON)
- PERMIT 'FEK.#CUST.PARMLIB' CLASS(DATASET) ACCESS(READ) ID(STCJMON)
- (オプション) クライアントに SCLMDT 用のロング/ショート・ネーム変換 VSAM 更新を許可する
 - PERMIT 'FEK.#CUST.LSTRANS*.**' CLASS(DATASET) ACCESS(UPDATE) ID(*)
- (オプション) RAM 開発者に CARMA 用の CARMA VSAM の更新を許可する
 - PERMIT 'FEK.#CUST.CRA*.**' CLASS(DATASET) ACCESS(UPDATE) ID(#ram-developer)
- (オプション) CICS ユーザーに、Application Deployment Manager 用の CRD リポジトリ VSAM の読み取りを許可します。
 - PERMIT 'FEK.#CUST.ADNREP*.**' CLASS(DATASET) ACCESS(READ) ID(*)
- (オプション) CICS 管理者に、Application Deployment Manager 用の CRD リポジトリ VSAM の更新を許可します。
 - PERMIT 'FEK.#CUST.ADNREP*.**' CLASS(DATASET) ACCESS(UPDATE) ID(#cicsadmin)
- (オプション) CICS ユーザーに、Application Deployment Manager 用のマニフェスト・リポジトリ VSAM の更新を許可する
 - PERMIT 'FEK.#CUST.ADNMAN*.**' CLASS(DATASET) ACCESS(UPDATE) ID(*)
- (オプション) CICS TS サーバーに bidi 用のロード・ライブラリーと Application Deployment Manager へのアクセスを許可する
 - PERMIT 'FEK.SFEKLOAD' CLASS(DATASET) ACCESS(READ) ID(#cicsts)
- (オプション) CICS TS サーバー、IMS 領域、および MVS バッチ・ジョブに、IRZ メッセージ用のロード・ライブラリーへのアクセスを許可する
 - PERMIT 'FEK.SFEKLMOD' CLASS(DATASET) ACCESS(READ) ID(#cicsts)
 - PERMIT 'FEK.SFEKLMOD' CLASS(DATASET) ACCESS(READ) ID(#ims)
 - PERMIT 'FEK.SFEKLMOD' CLASS(DATASET) ACCESS(READ) ID(#batch)
- セキュリティー・プロファイルをアクティブにする
 - SETROPTS GENERIC(DATASET) REFRESH

システム・データ・セットへの READ アクセスを制御するときは、Developer for System z サーバーおよびユーザーに以下のデータ・セットに対する READ 権限を与える必要があります。

- CEE.SCEERUN
- CEE.SCEERUN2
- CBC.SCLBDLL
- ISP.SISPLD
- ISP.SISPLPA
- SYS1.LINKLIB
- SYS1.SIEALNKE
- SYS1.SIEAMIGE
- REXX.V1R4M0.SEAGLPA

注: REXX 製品パッケージに代替ライブラリーを使用している場合、デフォルトの REXX ランタイム・ライブラリー名は、前のサンプルで使用されている REXX.*.SEAGLPA ではなく、REXX.*.SEAGALT となります。

セキュリティ設定の検査

セキュリティに関連するカスタマイズの結果を表示するには、以下のサンプル・コマンドを使用します。

- セキュリティの設定とクラス
 - SETROPTS LIST
- ユーザーの OMVS セグメント
 - LISTUSER #userid NORACF OMVS
 - LISTGRP #group-name NORACF OMVS
- 開始タスク
 - LISTGRP STCGROUP OMVS
 - LISTUSER STCDBM OMVS
 - LISTUSER STCJMON OMVS
 - LISTUSER STCRSE OMVS
 - RLIST STARTED DBGMR.* ALL STDATA
 - RLIST STARTED JMON.* ALL STDATA
 - RLIST STARTED RSED.* ALL STDATA
- セキュアな z/OS UNIX サーバーとしての RSE
 - RLIST FACILITY BPX.SERVER ALL
- RSE の MVS プログラム制御ライブラリー
 - RLIST PROGRAM ** ALL
- RSE の PassTicket サポート
 - RLIST PTKTDATA FEKAPPL ALL SSIGNON
 - RLIST PTKTDATA IRRPTAUTH.FEKAPPL.* ALL
- RSE のアプリケーション保護
 - RLIST APPL FEKAPPL ALL
- RSE の z/OS UNIX ファイル・アクセス許可
 - RLIST UNIXPRIV SUPERUSER.FILESYS ALL
 - RLIST UNIXPRIV SUPERUSER.FILESYS.CHOWN ALL
- JES コマンド・セキュリティ
 - RLIST CONSOLE JMON ALL
 - RLIST OPERCMDS MVS.MCSOPER.JMON ALL
 - RLIST OPERCMDS JES%.* ALL
- 統合デバッガーのアクセス
 - RLIST FACILITY AQE.* ALL
- データ・セット・プロファイル
 - LISTGRP FEK
 - LISTDSD PREFIX(FEK) ALL

オプションで、特定のユーザーに対する Developer for System z の動作を指定するプロファイルを使用できます。これらのプロファイルは FEK.** フィルターに一致

し、デフォルトで FACILITY クラスに置かれます。rsed.envvars の
_RSE_FEK_SAF_CLASS ディレクティブを確認してください。**SEARCH** コマンドを使用
してプロファイル名をリストすることができます。プロファイルの詳細を表示す
るには、**RLIST** コマンドを使用します。

- SEARCH CLASS(FACILITY) FILTER(FEK.**)
- RLIST FACILITY #profile-name ALL

第 10 章 マイグレーション・ガイド

マイグレーションに関する考慮事項

ここでは、本製品の以前のリリースと比較したインストールおよび構成上の変更点に重点を置いて説明します。また、このリリースへのマイグレーションに関する一般的なガイドラインも示します。詳しくは、本書の関連セクションを参照してください。

- 以前に IBM Rational Developer for System z、IBM WebSphere Developer for System z、IBM WebSphere Developer for zSeries、または IBM WebSphere® Studio Enterprise Developer を使用していた場合は、このバージョンの IBM Rational Developer for System z へのアップグレードを行う前に、関連するカスタマイズ済みのファイルを保存してください。
- Developer for System z の複数のインスタンスを実行する計画の場合は、「ホスト構成リファレンス」(SA88-4226) の『複数のインスタンスの実行』を参照してください。
- 3 つ以上離れたリリースにマイグレーションするシナリオの場合は、古い方のリリースは存在しないかのように、再びカスタマイズを行う必要があります。

前に構成したファイルのバックアップ

以前に Developer for System z を使用していた場合は、このバージョンの IBM Developer for System z をインストールする前に、関連するカスタマイズ済みのファイルを保存してください。

カスタマイズ可能な Developer for System z ファイルは、以下のロケーションにあります。

- バージョン 9.1、9.0、8.5、および 8.0.1
 - FEK.#CUST.RDZ*.*。構成ユーティリティの作業ファイル。
 - FEK.SFEKSAMP。一部のメンバーは、FEKSETUP サンプル・ジョブによって FEK.#CUST.* にコピーされます。ここで、* は PARMLIB、PROCLIB、JCL、CNTL、ASM、および COBOL に相当します。
 - FEK.SFEKSAMV
 - /usr/lpp/rdz/samples/。一部のファイルは、FEKSETUP サンプル・ジョブによって /etc/rdz/ および /var/rdz/scldmt/* にコピーされます。ここで、* は CONFIG/、CONFIG/PROJECT/、および CONFIG/script/ に相当します。

以前の Developer for System z セットアップでは、他の製品によって所有されている構成ファイルへの変更も記載されています。

- バージョン 9.1 および 9.0.1
 - SYS1.PARMLIB(IEASCVxx)

オプションの SVC を追加します。
- バージョン 9.0、8.5 および 8.0.1

- SYS1.PARMLIB(BPXPRMxx)
z/OS UNIX システムのデフォルトを設定します。
- SYS1.PARMLIB(COMMNDxx)
IPL 時にサーバーを始動します。
- SYS1.PARMLIB(LPALSTxx)
FEK.SFEKLPAを LPA に追加します。
- SYS1.PARMLIB(PROGxx)
FEK.SFEKAUTH および FEK.SFEKLOAD を LINKLIST に追加します。
- (WLM)
DB2 ストアード・プロシージャのアプリケーション環境を割り当てます。

バージョン 9.1 のマイグレーションに関する注意事項

以下のマイグレーションに関する注意事項は、IBM Rational Developer for System z バージョン 9.1 に固有のものです。これらの注意事項は、IBM Rational Developer for System z バージョン 9.1.0 からバージョン 9.1.1 へのマイグレーションに該当し、バージョン 9.1.0 のマイグレーションに関する既存の注意事項を補足するものです。

記載している変更内容はすべて、バージョン 9.1.1 以降に該当します。

IBM Rational Developer for System z、FMID HHOP910

- CARMA: CA Endeavor® SCM RAM 用の CRADEF ファイルおよび CRASTRS VSAM ファイルが更新されました。
- CARMA: 以下のカスタマイズ可能メンバーが変更されました。
 - CRACFG は新規オプション ALTERNATIVE-ALLOC を定義します。
 - CRABCFG は新規アクション TRANSFER-ELEMENT を定義します。
 - CRASUBCA に APIHJC が追加されました。
 - crastart.endevor.conf に APIHJC が追加されました。
- 統合デバッガー: IBM Debug Tool for z/OS に対する共存サポート
- 統合デバッガー: 以下のカスタマイズ可能メンバーが変更されました。
 - DBGMR 開始タスクは SVC 番号を使用しなくなりました。
 - AQECSD (CICS CSD 更新) は新規リソースを定義します。
 - AQERACF は新規 AQE.** プロファイルを定義します。
 - FEKRACF は新規 AQE.** プロファイルを定義します。
 - 統合デバッガー: 必須 CICS CSD 定義が追加されました。
 - AQEM および AQED 一時データ・キュー
 - プログラム
 - AQEW3Z3、AQEW3Z6、AQEL3Z3、AQEL3Z6、AQEL6Z3、AQEL6Z6
 - 統合デバッガー: 一部の CICS CSD 定義が削除されました。

- CIGZ および CIBM 一時データ・キュー
- プログラム AQEEV006
- 統合デバッガー: 必須 SVC が追加されました。
 - IEASVCxx parmlib が AQESVC03 用に変更されました。
- 統合デバッガー: オプションの SVC が削除されました。
 - IEASVCxx parmlib が AQESVC01 用に変更されました。
- 統合デバッガー: 必須 RACF プロファイルが追加されました。
 - AQE.AUTHDEBUG.STDPGM
 - AQE.AUTHDEBUG.AUTHPGM
- 統合デバッガー: オプション RACF プロファイルが削除されました。
 - AQE.AUTHDEBUG.WRITEBUFFER
- RSE: パスフレーズ・サポートが追加されました。
- RSE: オプション・ディレクティブが rsed.envvars に追加されました:
 - search.server.limit.scanned_objects
 - search.server.limit.errcount
- RSE: オプション・ディレクティブが rsed.envvars から削除されました:
 - enable.saf.check
 - RSE_DSICALL
 - search.server.limit.datasets

IBM Rational Developer for System z Host Utilities、FMID HAKG910

- コード・レビュー: AKGCR プロシーチャーの新規オプション
 - SYSLIB: クライアントからインポートされたデータを必要とする PROPERTY が置き換えられました。
 - LIST: データ・セット/メンバーと、それぞれのファイル・タイプのリスト (PDS、 MEMBERS、および EXTMAP が置き換えられました)
 - JUNIT: 新規出力形式

バージョン 9.0 からバージョン 9.1 へのマイグレーション

これらの注は、基本バージョン 9.0 からバージョン 9.1 へのマイグレーションのためのものです。これには、バージョン 9.0 のメンテナンスの一部として既に文書化されている変更点も含まれています。メンテナンス・ストリームの一部である、したがって既に実装されている可能性がある、変更内容には、それが導入された時点のリリースでマークが付いています。

既存の rsed.envvars (デフォルトでは /etc/rdz 内にあります) を、新たに提供されたサンプル (デフォルトでは /usr/lpp/rdz/samples 内にあります) で置き換えて、カスタマイズをやり直すことをお勧めします。

IBM Rational Developer for System z, FMID HHOP910

- SMP/E による MVS および z/OS UNIX コンポーネントのデフォルトのインストール・ロケーションは、変更されておらず、FEK.* および /usr/lpp/rdz/* のままです。
- CARMA: 以下のカスタマイズ可能メンバーが変更されました。
 - CRASUBMT
 - CRASUBCA
- カスタマイズ: FEKSETUP JCL が新規メンバーを処理するようになりました。
 - AQED3CEE: FEK.#CUST.JCL(AQED3CEE) にコピーされる
 - AQED3CXT: FEK.#CUST.JCL(AQED3CXT) にコピーされる
 - FEKPBITS: FEK.#CUST.JCL(FEKPBITS) にコピーされる
- RSE: 以下の新しいオプション・ディレクティブが rsed.envvars に追加されました。
 - (_RSE_JAVAOPTS) -Denable.ddvipa
 - (_RSE_JAVAOPTS) -Dlog.file.mode
 - (_RSE_JAVAOPTS) -Dlog.secure.mode
 - (_RSE_JAVAOPTS) -Denable.saf.check
 - (_RSE_JAVAOPTS) -Dkeep.stats.copy.local
 - (_RSE_JAVAOPTS) -Ddebug.miner.localhost
 - (_RSE_JAVAOPTS) -DDSTORE_USE_THREADED_MINERS
- RSE: 以下の新しいオペレーター・コマンドが追加されました。
 - F rsed,APPL=LOG {USER | AUDIT | NOSERVER | OWNER}
- RSE: 以下の新しいコンソール・メッセージが追加されました。
 - FEK220I = ホスト・ログは {0} に書き込まれます (Host logs are written onto {0})
 - FEK221E = {0} は {1} が原因で中断しました ({0} was interrupted because of {1})
 - FEK301E = {0} (uid:{1}) は {2} (file_owner uid:{3}) のディレクトリーを所有しません ({0} (uid:{1}) does not own the directory of {2} (file_owner uid:{3}))
 - FEK302E = LOGS コマンドの要求者 {0} には、{1} のプロファイルにアクセスする権限がありません (The requester, {0}, of the LOGS command does not have authority to access the profile of {1})
 - FEK303E = シンボリック・リンク {0} はログ・ディレクトリーとして使用できません (The symbolic link, {0}, cannot be used as a log directory)
 - FEK304W = 無効な {0}、{1} が指定されました。代わりに、デフォルト・モード {3} が使用されます。 (Invalid {0}, {1}, was specified. The default mode, {3}, is used instead.)
 - FEK305E = ID {0} は、{1} にアクセスするための適切な特権を保持していません。 (The ID, {0}, does not have appropriate privileges to access {1}.)
- セキュリティー: 以下のセキュア・ログ・ファイルに対するサポートが追加されました。

- FEKPBITS: 既存のログ・インフラストラクチャーの許可を変更するためのスクリプト
- 新規動作: サーバー・ログが `$daemon.log/server` に配置されるようになりました。
- 新規動作: ログは制限付きアクセス許可で作成されます。
- セキュリティー: 以下の新しいセキュリティー・プロファイルのサポートが追加されました。
 - FEK.CMD.LOGS.**
 - FEK.REJECT.*.UPDATES.system.group
- バージョン 9.0.1.1 以降
 - RSE: 以下の新しいオプション・ディレクティブが `rsed.envvars` に追加されました。
 - (`_RSE_JAVAOPTS`) -Ddebug.miner.autoreconnect
- バージョン 9.0.1 以降
 - CARMA: CA Endeavor® SCM RAM 用の CRADEF VSAM ファイルが更新されました。
 - CARMA: LPA 内にある CRASTART ロード・モジュールは更新されており、LPA の更新が必要です。
 - CARMA: CARMA 始動時にユーザー出口を実行するためのサポートが追加されました。
 - CARMA: RAM 処理始動引数のサポートが追加されました。
 - CARMA: 以下の新しいカスタマイズ可能メンバーが追加されました。
 - CRAEXIT: サンプル CARMA ユーザー出口。
 - CRAALLOC: カスタム RAM CARMA 呼び出し用の割り振り exec。
 - CRACFG: CA Endeavor® SCM RAM 使用法構成ファイル。
 - CARMA: 以下のカスタマイズ可能メンバーが変更されました。
 - CRASRV.properties
 - crastart.conf
 - crastart.endevor.conf
 - CRASUBMT
 - CRASUBCA
 - CRANDVRA
 - CARMA: crastart.endevor.conf および CRASUBCA の CA Endeavor® SCM RAM に対して DD ステートメントがさらに追加されました。
 - CRAPARM (CRANDVRA により割り振られる)
 - CRACFG
 - CARMA: crastart.conf および CRASUBMT の CA Endeavor® SCM RAM 以外のものに対して DD ステートメントがさらに追加されました。
 - CRAPARM (CRAALLOC により割り振られる)
 - カスタマイズ: FEKSETUP JCL が新規メンバーを処理するようになりました。
 - CRACFG: FEK.#CUST.PARMLIB(CRACFG) にコピーされる

- AQEJCL: FEK.#CUST.PROCLIB(DBGMGR) にコピーされる
- AQECSD: FEK.#CUST.JCL(AQECSD) にコピーされる
- 統合デバッガー: 新しいオプション・サービス
 - IEASVCxx、LPALSTxx、および PROGxx (APF および LINKLIST) PARMLIB 更新
 - DBGMGR: 開始タスク JCL
 - AQECSD: CICS CSD を更新するサンプル JCL
 - AQERACF: 統合デバッガーのみにセキュリティ・セットアップを実行するサンプル JCL
- RSE: 更新された PROCLIB メンバー
 - ELAXFGO
- RSE: 以下の新しいオプション・ディレクティブが rsecomm.properties に追加されました。
 - USER
- RSE: 新しいオペレーター・コマンド
 - F rsed,APPL=TRACE {USER | SERVER | CLEAR}
- RSE: 以下の新しいオプション・ディレクティブが rsed.envvars に追加されました。
 - (_RSE_JAVAOPTS) -Dsearch.server.limit.timeout
 - (_RSE_JAVAOPTS) -Dkeep.all.logs
 - (_RSE_JAVAOPTS) -Daudit.users
 - RSE_UBLD_DD
 - RSE_UBLD_STEPLIB
- RSE: 以下の新しいコンソール・メッセージが追加されました。
 - FEK910I = FEK107E = {0} のディスクがほとんど満杯です。既存の {1} が削除されました
- zUnit: 以下の新しいオプションの始動引数が追加されました。
 - CLOCALE / -l

注: 統合デバッガーがない状態の、既存の Developer for System z セットアップからのマイグレーションを単純化するために、RACF コマンドを伴うサンプル JCL FEK.SFEKSAMP(AQERACF) が提供されています。これは統合デバッガーに関連するセキュリティ定義だけを定義します。

構成可能なファイル

195 ページの表 24 は、バージョン 9.1.0 でカスタマイズされるファイルの概要を示しています。Developer for System z のサンプル・ライブラリー FEK.SFEKSAMP、FEK.SFEKSAMV、および /usr/lpp/rdz/samples/ には、ここに示したものより多くのカスタマイズ可能なメンバー (サンプルの CARMA ソース・コードおよびそれらをコンパイルするジョブなど) が含まれています。

注: サンプル・ジョブ FEKSETUP は、リストされているすべてのメンバーを別のデータ・セットおよびディレクトリー (デフォルトでは FEK.#CUST.* および /etc/rdz/*) にコピーします。

表 24. バージョン 9.1.0 のカスタマイズ

メンバー/ファイル	デフォルト・ロケーション	目的	マイグレーションの注
FEKSETUP	FEK.SFEKSAMP [FEK.#CUST.JCL]	データ・セットおよびディレクトリーを作成し、カスタマイズ可能ファイルのデータを取り込むための JCL	使用されなくなったファイルに対するアクションを削除し、新規ファイル用のアクションを追加するために更新
JMON	FEK.SFEKSAMP(FEJJJCL) [FEK.#CUST.PROCLIB]	JES ジョブ・モニター用の JCL	なし
FEJJJCL	FEK.SFEKSAMP [FEK.#CUST.PROCLIB(JMON)]	JMON メンバーの名前	JMON メンバーを参照
RSED	FEK.SFEKSAMP(FEKRSED) [FEK.#CUST.PROCLIB]	RSE デーモンの JCL	なし
FEKRSED	FEK.SFEKSAMP [FEK.#CUST.PROCLIB(RSED)]	RSED メンバーの名前	RSED メンバーを参照
DBGMGR	FEK.SFEKSAMP [FEK.#CUST.JCL]	デバッグ・マネージャー用の JCL	新規、カスタマイズはオプション
AQEJCL	FEK.SFEKSAMP [FEK.#CUST.PROCLIB(DBGMGR)]	DBGMGR メンバーの名前	DBGMBR メンバーを参照
ELAXF*	FEK.SFEKSAMP [FEK.#CUST.PROCLIB]	リモート・プロジェクト・ビルドなどのための JCL	ELAXFGO には STEPLIB の SFEKAUTH が含まれる
FEKRACF	FEK.SFEKSAMP [FEK.#CUST.JCL]	セキュリティ定義の JCL	新規、開始タスク DBGMGR
AQERACF	FEK.SFEKSAMP [FEK.#CUST.JCL]	デバッグ・マネージャー用の JCL	新規、カスタマイズはオプション
FEKPBITS	FEK.SFEKSAMP [FEK.#CUST.JCL]	ログ・ファイルのアクセス許可を変更するための JCL	新規、カスタマイズはオプション
FEJJCNG	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	JES ジョブ・モニター構成ファイル	なし
FEJTSO	FEK.SFEKSAMP [FEK.#CUST.CNTL]	TSO 実行依頼用の JCL	なし

表 24. バージョン 9.1.0 のカスタマイズ (続き)

メンバー/ファイル	デフォルト・ロケーション	目的	マイグレーションの注
CRA\$VMMSG	FEK.SFEKSAMP [FEK.#CUST.JCL]	CARMA メッセージ VSAM を作成するための JCL	なし
CRA\$VDEF	FEK.SFEKSAMP [FEK.#CUST.JCL]	CARMA 構成 VSAM を作成するための JCL	なし
CRA\$VSTR	FEK.SFEKSAMP [FEK.#CUST.JCL]	CARMA カスタム情報 VSAM を作成するための JCL	なし
CRA\$VCAD	FEK.SFEKSAMP [FEK.#CUST.JCL]	CA Endeavor [®] SCM RAM 用に CARMA 構成 VSAM を作成するための JCL	VSAM 入力に変更された
CRA\$VCAS	FEK.SFEKSAMP [FEK.#CUST.JCL]	CA Endeavor [®] SCM RAM 用に CARMA カスタム情報 VSAM を作成するための JCL	VSAM 入力に変更された
CRASUBMT	FEK.SFEKSAMP [FEK.#CUST.CNTL]	CARMA バッチ始動 CLIST	8 つの始動引数、DD CRAPARM および割り振り exec のサポートを追加
CRASUBCA	FEK.SFEKSAMP [FEK.#CUST.CNTL]	CA Endeavor [®] SCM RAM 用の CARMA バッチ始動 CLIST	8 つの始動引数、DD CRAPARM および新規 DD のサポートを追加
CRACFG	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	CA Endeavor [®] SCM RAM 用の CARMA 対話構成	新規、構成はオプション

表 24. バージョン 9.1.0 のカスタマイズ (続き)

メンバー/ファイル	デフォルト・ロケーション	目的	マイグレーションの注
CRABCFG	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	CA Endeavor® SCM RAM 用の CARMA バッチ・アクション構成	なし
CRABATCA	FEK.SFEKSAMP [FEK.#CUST.CNTL]	CA Endeavor® SCM RAM 用の CARMA バッチ・アクションの JCL	なし
CRASHOW	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	CA Endeavor® SCM RAM 用の CARMA 構成	なし
CRATMAP	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	CA Endeavor® SCM RAM 用の CARMA 構成	なし
CRANDVRA	FEK.SFEKPROC	CA Endeavor® SCM RAM 用の CARMA 割り振り REXX	ユーザー出口のサポートが追加され、新規 DD を割り振る
CRAALLOC	FEK.SFEKPROC	CARMA 割り振り REXX	新規、構成はオプション
CRA#VSLM	FEK.SFEKSAMP [FEK.#CUST.JCL]	SCLM RAM のメッセージ VSAM を作成するための JCL	なし
CRA#ASLM	FEK.SFEKSAMP [FEK.#CUST.JCL]	SCLM RAM のデータ・セットを作成するための JCL	なし
CRA#VPDS	FEK.SFEKSAMP [FEK.#CUST.JCL]	PDS RAM のメッセージ VSAM を作成するための JCL	なし
CRA#UADD	FEK.SFEKSAMP [FEK.#CUST.JCL]	RAM 定義をマージするための JCL	なし
CRA#UQRY	FEK.SFEKSAMP [FEK.#CUST.JCL]	RAM 定義を抽出するための JCL	なし

表 24. バージョン 9.1.0 のカスタマイズ (続き)

メンバー/ファイル	デフォルト・ロケーション	目的	マイグレーションの注
CRAXJCL	FEK.SFEKSAMP [FEK.#CUST.ASM]	IRXJCL 置換 用のサンプ ル・ソース・ コード	なし
CRA#CIRX	FEK.SFEKSAMP [FEK.#CUST.JCL]	CRAXJCL を コンパイルす るための JCL	なし
AQECSD	FEK.SFEKSAMP [FEK.#CUST.JCL]	統合デバッガ ーを CICS 領 域に対して定 義するための JCL	新規、構成は オプション
AQED3CEE	FEK.SFEKSAMP [FEK.#CUST.JCL]	カスタマイズ された LE ラ ンタイム・モ ジュールを作 成するための JCL	新規、構成は オプション
AQED3CXT	FEK.SFEKSAMP [FEK.#CUST.JCL]	LE ユーザー 出口を作成す るための JCL	新規、構成は オプション
ADNCSDRS	FEK.SFEKSAMP [FEK.#CUST.JCL]	RESTful CRD サーバーを主 CICS 領域に 対して定義す るための JCL	なし
ADNCSDTX	FEK.SFEKSAMP [FEK.#CUST.JCL]	代替トランザ クション ID を CICS 領域 に対して定義 するための JCL	なし
ADNTXNC	FEK.SFEKSAMP [FEK.#CUST.JCL]	代替トランザ クション ID を作成するた めの JCL	なし
ADNMSGHC	FEK.SFEKSAMP [FEK.#CUST.JCL]	ADNMSGHS をコンパイル するための JCL	なし
ADNMSGHS	FEK.SFEKSAMP [FEK.#CUST.COBOL]	パイプライ ン・メッセー ジ・ハンドラ ー用のサンプ ル・ソース・ コード	なし

表 24. バージョン 9.1.0 のカスタマイズ (続き)

メンバー/ファイル	デフォルト・ロケーション	目的	マイグレーションの注
ADNVCRD	FEK.SFEKSAMP [FEK.#CUST.JCL]	CRD リポジトリを作成するための JCL	なし
ADNCSDWS	FEK.SFEKSAMP [FEK.#CUST.JCL]	Web サービス CRD サーバーを主 CICS 領域に対して定義するための JCL	なし
ADNCSDAR	FEK.SFEKSAMP [FEK.#CUST.JCL]	CRD サーバーを非主 CICS 領域に対して定義するための JCL	なし
ADNJSPAU	FEK.SFEKSAMP [FEK.#CUST.JCL]	CRD のデフォルトを更新するための JCL	なし
ADNVMFST	FEK.SFEKSAMP [FEK.#CUST.JCL]	マニフェスト・リポジトリを作成し、定義するための JCL	なし
FEKTEP2	FEK.SFEKSAMP [FEK.#CUST.SQL]	ELAXF* で使用される SQL コマンド入力	なし
FEKTIAD	FEK.SFEKSAMP [FEK.#CUST.JCL]	ELAXF* で使用される SQL コマンド入力	なし
AZUZUNIT	FEK.SFEKSAMP [FEK.#CUST.PROCLIB]	zUnit プロシージャの JCL	なし
FEKRNPLI	FEK.SFEKSAMP [FEK.#CUST.CNTL]	プリプロセッサ・フレームワーク内部から PL/I コンパイラーを呼び出すための REXX	なし
FEKLOGS	FEK.SFEKSAMP [FEK.#CUST.JCL]	ログ・ファイルを収集するための JCL	なし

表 24. バージョン 9.1.0 のカスタマイズ (続き)

メンバー/ファイル	デフォルト・ロケーション	目的	マイグレーションの注
rsed.envvars	/usr/lpp/rdz/samples/ [/etc/rdz/]	RSE 環境変数	古いコピーをこのコピーに置き換え、カスタマイズを再実行する必要がある。
ISPF.conf	/usr/lpp/rdz/samples/ [/etc/rdz/]	TSO/ISPF クライアント・ゲートウェイ構成ファイル	なし
CRASRV.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	CARMA 構成ファイル	ユーザー出口のサポートを追加
crastart.conf	/usr/lpp/rdz/samples/ [/etc/rdz/]	CRASTART を使用するための CARMA 構成ファイル	8 つの始動引数、DD CRAPARM および割り振り exec のサポートを追加
crastart.endevor.conf	/usr/lpp/rdz/samples/ [/etc/rdz/]	CA Endeavor [®] SCM RAM 用に CRASTART を使用するための CARMA 構成ファイル	8 つの始動引数、DD CRAPARM および新規 DD のサポートを追加
include.conf	/usr/lpp/rdz/samples/ [/etc/rdz/]	C/C++ コンテンツ・アシスト用の強制インクルード	なし
ssl.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	RSE SSL 構成ファイル	なし
rsecomm.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	RSE トレース構成ファイル	USER キーワードのサポートを追加
pushtoclient.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	クライアント構成ファイルに情報をプッシュする	なし

IBM Rational Developer for System z Host Utilities、FMID HAKG910

- SMP/E による MVS および z/OS UNIX コンポーネントのデフォルトのインストール・ロケーションは変更されておらず、したがって AKG.* および /usr/lpp/rdzutil/* のままです。
- コード・レビュー: 追加の DD ステートメントが AKGCR に追加されました。
 - BEXPORT
 - BIMPORT

構成可能なファイル

表 25 は、バージョン 9.1 でカスタマイズされるファイルの概要を示しています。Developer for System z Host Utilities のサンプル・ライブラリーの AKG.SAKGSAMP および /usr/lpp/rdzutil/samples には、ここに示したものより多くのカスタマイズ可能なメンバー (サンプルのコード・レビューの後処理スクリプトなど) が含まれます。

注: サンプル・ジョブ AKGSETUP は、リストされているすべてのメンバーを別のデータ・セット (デフォルトでは AKG.#CUST.*) にコピーします。

表 25. Host Utilities バージョン 9.0 のカスタマイズ

メンバーまたはファイル	デフォルト・ロケーション	目的	マイグレーションの注
AKGSETUP	AKG.SAKGSAMP [AKG.#CUST.JCL]	データ・セットを作成し、カスタマイズ可能ファイルのデータを取り込むための JCL	なし
AKGCC	AKG.SAKGSAMP [AKG.#CUST.PROCLIB]	コード・カバレッジの JCL	なし
AKGCR	AKG.SAKGSAMP [AKG.#CUST.PROCLIB]	コード・レビューの JCL	新規 DD BIMPORT および BEXPORT
AKGCRADD	AKG.SAKGSAMP [AKG.#CUST.JCL]	サード・パーティーのコードをコード・レビューに追加するための JCL	なし

バージョン 8.5 からバージョン 9.0 へのマイグレーション

これらの注は、基本バージョン 8.5 からバージョン 9.0 へのマイグレーションのためのものです。これには、バージョン 8.5 のメンテナンスの一部として既に文書化されている変更点も含まれています。メンテナンス・ストリームの一部である、したがって既に実装されている可能性がある、変更内容には、それが導入された時点のリリースでマークが付いています。

IBM Rational Developer for System z, FMID HHOP900

- SMP/E による MVS および z/OS UNIX コンポーネントのデフォルトのインストール・ロケーションは、変更されておらず、したがって FEK.* および /usr/lpp/rdz/* のままです。
- CARMA: カスタマイズ可能な CA Endevor® SCM バッチ・アクション (バージョン 8.5.1 より) に対する新しいサポートを使用するには、CA Endevor® SCM RAM 用の CRADEF ファイルおよび CRASTRS VSAM ファイルを更新する必要があります。
- CARMA: CRADEF VSAM 作成時に RAM を使用不可にするサポートが追加されました (バージョン 8.5.1 以降)。
- CARMA: CRASRV.properties の相対ファイル参照のサポートが追加されました (バージョン 8.5.1 以降)。
- CARMA: 新しいサンプル・メンバーが追加されました。
 - CRABJOBC: CA Endevor® SCM バッチ・アクション用のデフォルト JOB カード (バージョン 8.5.1 より)。
- CARMA: 以下のカスタマイズ可能メンバーが変更されました。
 - CRASRV.properties (バージョン 8.5.1 以降)
 - carma.startup.rex (バージョン 8.5.1 以降)
 - CRA\$VCAD (バージョン 8.5.1 以降)
 - CRA\$VDEF (バージョン 8.5.1 以降)
 - CRABATCA (バージョン 8.5.1 以降)
 - CRABCFG (バージョン 8.5.1 以降)
 - CRANDVRA (バージョン 8.5.1 以降)
- CARMA: crastart.endevor.conf および CRASUBCA の CA Endevor® SCM RAM に対して DD ステートメントがさらに追加されました。
 - CRABJCLO (CRANDVRA により割り振られる) (バージョン 8.5.1 以降)
 - ENHCEDIT (CRANDVRA により割り振られる) (バージョン 8.5.1 以降)
- カスタマイズ: FEKSETUP JCL が新規メンバーを処理するようになりました。
 - CRABJOBC: FEK.#CUST.CNTL(CRABJOBC) にコピーされます (バージョン 8.5.1 以降)
 - ELAXFSP: FEK.#CUST.PROCLIB(ELAXFSP) にコピーされます (バージョン 9.0 以降)
 - ELAXFSQL: FEK.#CUST.PROCLIB(ELAXFSQL) にコピーされます (バージョン 9.0 以降)
 - FEKTEP2: FEK.#CUST.SQL(FEKTEP2) にコピーされます (バージョン 9.0 以降)
 - FEKTIAD: FEK.#CUST.SQL(FEKTEP2) にコピーされます (バージョン 9.0 以降)
- Fault Analyzer 統合: FAI のサポートは廃止されました。この変更は、まだ FAI を使用している古いクライアントとは非互換です。
- JES ジョブ・モニター - JMON 開始タスクに対して新しいオペレーター・コマンドが追加されました。

- MODIFY USERS (バージョン 8.5.1 以降)
- MODIFY -T{N | E | I | V} (バージョン 8.5.1 以降)
- MODIFY -M{N | E | W | I | V} (バージョン 8.5.1 以降)
- MODIFY TRACE {N | E | I | V} (バージョン 9.0 以降)
- MODIFY MESSAGE {N | E | W | I | V} (バージョン 9.0 以降)
- JES ジョブ・モニター - 以下の新しいオプション・ディレクティブが FEJCNFG に追加されました。
 - LOOPBACK_ONLY (バージョン 9.0 以降)
- JES ジョブ・モニター - 以下のオプション・ディレクティブが FEJCNFG から削除されました。
 - _BPXK_SETIBMOPT_TRANSPORT (バージョン 9.0 以降)
- 問題判別: FEKLOGS JCL は、ユーザー・ログを収集するために、複数のユーザー ID の指定をサポートするようになりました (バージョン 8.5.1 より)。
- 問題判別: FEKLOGS JCL は DD REFORMAT を使用して再フォーマットされたログを収集し、より迅速に問題を判別します (バージョン 8.5.1 より)。
- 問題判別: 以下のカスタマイズ可能メンバーが変更されました。
 - FEKLOGS (バージョン 8.5.1 以降)
- RSE - 以下の新しいオペレーター・コマンドが RSED 開始タスクに追加されました。
 - MODIFY DISPLAY OWNER,DATASET=dataset (バージョン 9.0 以降)
 - MODIFY DEBUG GC,PID=pid (バージョン 9.0 以降)
- RSE: 以下のカスタマイズ不可の新しいディレクティブが rsed.envvars に追加されました。
 - _CMDSESV_BASE_HOME (バージョン 8.5.1 以降)
 - _CMDSESV_CONF_HOME (バージョン 8.5.1 以降)
 - _CMDSESV_WORK_HOME (バージョン 8.5.1 以降)
 - RSE_DSN_SFELLOAD (バージョン 9.0 以降)
 - (_RSE_JAVAOPTS) -Dlock.info.timeout (バージョン 9.0 以降)
 - (_RSE_JAVAOPTS) -DDSTORE_INITIAL_SIZE (バージョン 9.0 以降)
 - (_RSE_JAVAOPTS) -DDSTORE_MAX_FREE (バージョン 9.0 以降)
- RSE: 以下の新しい必須のディレクティブが rsed.envvars に追加されました。
 - RSE_HLQ (バージョン 9.0 以降)
- RSE: 以下の新しいオプション・ディレクティブが rsed.envvars に追加されました。
 - (_RSE_JAVAOPTS) -DRSE_DSICALL (バージョン 8.5.1 以降)
 - (_RSE_JAVAOPTS) -DDISABLE_REMOTE_INDEX_SEARCH (バージョン 8.5.1 以降)
 - (_RSE_JAVAOPTS) -DDISABLE_TEXT_SEARCH (バージョン 9.0 以降)
 - (_RSE_JAVAOPTS) -Dsearch.server.limit.hits (バージョン 9.0 以降)
 - (_RSE_JAVAOPTS) -Dsearch.server.limit.datasets (バージョン 9.0 以降)
 - (_RSE_JAVAOPTS) -Dsearch.server.limit.lines (バージョン 9.0 以降)

- (_RSE_JAVAOPTS) -DDSTORE_SSL_ALGORITHM (バージョン 9.0 以降)
- RSE: rsed.envvars の以下のカスタマイズ不可のディレクティブのデフォルト値が変更されました。
 - (_RSE_JAVAOPTS) -DSPIRIT_EXPIRY_TIME (バージョン 9.0 以降)
- RSE: rsed.envvars の以下のオプション・ディレクティブのデフォルト値が変更されました。
 - (_RSE_JAVAOPTS) -Xms (バージョン 8.5.1 以降)
 - (_RSE_JAVAOPTS) -Xmx (バージョン 8.5.1 以降)
 - (_RSE_JAVAOPTS) -Dmaximum.clients (バージョン 8.5.1 以降)
 - (_RSE_JAVAOPTS) -Dmaximum.threads (バージョン 8.5.1 以降)
 - CGI_ISPPREF (バージョン 9.0 以降)
- セキュリティー: 以下の新しいセキュリティー・プロファイルのサポートが追加されました。
 - FEK.USR.** (バージョン 8.5.1 以降)

構成可能なファイル

表 26 は、バージョン 9.0 でカスタマイズされるファイルの概要を示しています。Developer for System z のサンプル・ライブラリー FEK.SFEKSAMP、FEK.SFEKSAMV、および /usr/lpp/rdz/samples/ には、ここに示したものより多くのカスタマイズ可能なメンバー (サンプルの CARMA ソース・コードおよびそれらをコンパイルするジョブなど) が含まれています。

以下のメンバーおよびファイルはもうカスタマイズ可能でないか、使用されなくなっています。

- LOCKD 開始タスク
- ELAXMSAM サンプル DB2 ストアード・プロシージャ
- DB2 ストアード・プロシージャ用の ELAXMJCL サンプル JCL

注: サンプル・ジョブ FEKSETUP は、リストされているすべてのメンバーを別のデータ・セットおよびディレクトリー (デフォルトでは FEK.#CUST.* および /etc/rdz/*) にコピーします。

表 26. バージョン 9.0 のカスタマイズ

メンバー/ファイル	デフォルト・ロケーション	目的	マイグレーションの注
FEKSETUP	FEK.SFEKSAMP [FEK.#CUST.JCL]	データ・セットおよびディレクトリーを作成し、カスタマイズ可能ファイルのデータを取り込むための JCL	使用されなくなったファイルに対するアクションを削除し、新規ファイル用のアクションを追加するために更新

表 26. バージョン 9.0 のカスタマイズ (続き)

メンバー/ファイル	デフォルト・ロケーション	目的	マイグレーションの注
JMON	FEK.SFEKSAMP (FEJJJCL) [FEK.#CUST.PROCLIB]	JES ジョ ブ・モニター 用の JCL	なし
FEJJJCL	FEK.SFEKSAMP [FEK.#CUST.PROCLIB(JMON)]	JMON メン バーの名前	JMON メンバ ーを参照
RSED	FEK.SFEKSAMP (FEKRSED) [FEK.#CUST.PROCLIB]	RSE デーモ ンの JCL	なし
FEKRSED	FEK.SFEKSAMP [FEK.#CUST.PROCLIB(RSED)]	RSED メン バーの名前	RSED メンバ ーを参照
ELAXF*	FEK.SFEKSAMP [FEK.#CUST.PROCLIB]	リモート・プ ロジェクト・ ビルドなどの ための JCL	ELAXFSP およ び ELAXFSQL が追加され、 ELAXFCOC お よび ELAXFCP1 は COBOL パー ジョン 5 をサ ポートするよ うに更新され た
FEKRACF	FEK.SFEKSAMP [FEK.#CUST.JCL]	セキュリティ 一定義の JCL	なし
FEJJCNGF	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	JES ジョ ブ・モニター 構成ファイル	オプションの 新規ディレク ティブが追加 された。既存 のオプション のディレクテ ィブは削除さ れた。
FEJT50	FEK.SFEKSAMP [FEK.#CUST.CNTL]	TSO 実行依 頼用の JCL	なし
CRA\$VMSG	FEK.SFEKSAMP [FEK.#CUST.JCL]	CARMA メ ッセージ VSAM を作 成するための JCL	なし
CRA\$VDEF	FEK.SFEKSAMP [FEK.#CUST.JCL]	CARMA 構 成 VSAM を 作成するた めの JCL	RAM を除外す るためのサポ ートが追加さ れた

表 26. バージョン 9.0 のカスタマイズ (続き)

メンバー/ファイル	デフォルト・ロケーション	目的	マイグレーションの注
CRA\$VSTR	FEK.SFEKSAMP [FEK.#CUST.JCL]	CARMA カスタム情報 VSAM を作成するための JCL	なし
CRA\$VCAD	FEK.SFEKSAMP [FEK.#CUST.JCL]	CA Endeavor [®] SCM RAM 用に CARMA 構成 VSAM を 作成するための JCL	RAM を除外する ためのサポートが追加 され、VSAM 入力 が変更された。
CRA\$VCAS	FEK.SFEKSAMP [FEK.#CUST.JCL]	CA Endeavor [®] SCM RAM 用に CARMA カスタム情報 VSAM を作成するための JCL	VSAM 入力 が変更された
CRASUBMT	FEK.SFEKSAMP [FEK.#CUST.CNTL]	CARMA バッチ始動 CLIST	なし
CRASUBCA	FEK.SFEKSAMP [FEK.#CUST.CNTL]	CA Endeavor [®] SCM RAM 用の CARMA バッチ始動 CLIST	なし
CRABCFG	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	CA Endeavor [®] SCM RAM 用の CARMA バッチ・アクシ ョン構成	新しいディレ クティブが追加 された
CRABATCA	FEK.SFEKSAMP [FEK.#CUST.CNTL]	CA Endeavor [®] SCM RAM 用の CARMA バッチ・アクシ ョンの JCL	変数ジョブ・ カードのサポ ートが追加 された
CRASHOW	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	CA Endeavor [®] SCM RAM 用の CARMA 構成	なし

表 26. バージョン 9.0 のカスタマイズ (続き)

メンバー/ファイル	デフォルト・ロケーション	目的	マイグレーションの注
CRATMAP	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	CA Endeavor® SCM RAM 用の CARMA 構 成	なし
CRANDVRA	FEK.SFEKPROC	CA Endeavor® SCM RAM 用の CARMA 割 り振り REXX	新しい DD 割 り振りが追加 された
CRA#VSLM	FEK.SFEKSAMP [FEK.#CUST.JCL]	SCLM RAM のメッセージ VSAM を作 成するための JCL	なし
CRA#ASLM	FEK.SFEKSAMP [FEK.#CUST.JCL]	SCLM RAM のデータ・セ ットを作成す るための JCL	なし
CRA#VPDS	FEK.SFEKSAMP [FEK.#CUST.JCL]	PDS RAM のメッセージ VSAM を作 成するための JCL	なし
CRA#UADD	FEK.SFEKSAMP [FEK.#CUST.JCL]	RAM 定義を マージするた めの JCL	なし
CRA#UQRY	FEK.SFEKSAMP [FEK.#CUST.JCL]	RAM 定義を 抽出するた めの JCL	なし
CRAXJCL	FEK.SFEKSAMP [FEK.#CUST.ASM]	IRXJCL 置換 用のサンプ ル・ソース・ コード	なし
CRA#CIRX	FEK.SFEKSAMP [FEK.#CUST.JCL]	CRAXJCL を コンパイルす るための JCL	なし
ADNCSDRS	FEK.SFEKSAMP [FEK.#CUST.JCL]	RESTful CRD サーバ ーを主 CICS 領域に対して 定義するた めの JCL	なし

表 26. バージョン 9.0 のカスタマイズ (続き)

メンバー/ファイル	デフォルト・ロケーション	目的	マイグレーションの注
ADNCSDTX	FEK.SFEKSAMP [FEK.#CUST.JCL]	代替トランザクション ID を CICS 領域に対して定義するための JCL	なし
ADNTXNC	FEK.SFEKSAMP [FEK.#CUST.JCL]	代替トランザクション ID を作成するための JCL	なし
ADNMSGHC	FEK.SFEKSAMP [FEK.#CUST.JCL]	ADNMSGHS をコンパイルするための JCL	なし
ADNMSGHS	FEK.SFEKSAMP [FEK.#CUST.COBOL]	パイプライン・メッセージ・ハンドラー用のサンプル・ソース・コード	なし
ADNVCRD	FEK.SFEKSAMP [FEK.#CUST.JCL]	CRD リポジトリを作成するための JCL	なし
ADNCSDWS	FEK.SFEKSAMP [FEK.#CUST.JCL]	Web サービス CRD サーバーを主 CICS 領域に対して定義するための JCL	なし
ADNCSDAR	FEK.SFEKSAMP [FEK.#CUST.JCL]	CRD サーバーを非主 CICS 領域に対して定義するための JCL	なし
ADNJSPAU	FEK.SFEKSAMP [FEK.#CUST.JCL]	CRD のデフォルトを更新するための JCL	なし
ADNVMFST	FEK.SFEKSAMP [FEK.#CUST.JCL]	マニフェスト・リポジトリを作成し、定義するための JCL	なし

表 26. バージョン 9.0 のカスタマイズ (続き)

メンバー/ファイル	デフォルト・ロケーション	目的	マイグレーションの注
FEKTEP2	FEK.SFEKSAMP [FEK.#CUST.SQL]	ELAXF* で 使用される SQL コマン ド入力	新規、カスタ マイズはオブ ション
FEKTIAD	FEK.SFEKSAMP [FEK.#CUST.JCL]	ELAXF* で 使用される SQL コマン ド入力	新規、カスタ マイズはオブ ション
AZUZUNIT	FEK.SFEKSAMP [FEK.#CUST.PROCLIB]	zUnit プロシ ージャの JCL	なし
FEKRNPLI	FEK.SFEKSAMP [FEK.#CUST.CNTL]	プリプロセッ サー・フレー ムワーク内部 から PL/I コ ンパイラーを 呼び出すため の REXX	なし
FEKLOGS	FEK.SFEKSAMP [FEK.#CUST.JCL]	ログ・ファイ ルを収集する ための JCL	その他の検査 が追加され た。古いファ イルに加えた カスタマイズ はすべて再実 行する必要が ある。
rsed.envvars	/usr/lpp/rdz/samples/ [/etc/rdz/]	RSE 環境変 数	古いコピーを このコピーに 置き換え、カ スタマイズを 再実行する必 要がある。
ISPF.conf	/usr/lpp/rdz/samples/ [/etc/rdz/]	TSO/ISPF ク ライアント・ ゲートウェイ 構成ファイル	なし
CRASRV.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	CARMA 構 成ファイル	デフォルト値 のサポートを 追加
crastart.conf	/usr/lpp/rdz/samples/ [/etc/rdz/]	CRASTART を使用するた めの CARMA 構 成ファイル	なし

表 26. バージョン 9.0 のカスタマイズ (続き)

メンバー/ファイル	デフォルト・ロケーション	目的	マイグレーションの注
crastart.endevor.conf	/usr/lpp/rdz/samples/ [/etc/rdz/]	CA Endevor® SCM RAM 用に CRASTART を使用するた めの CARMA 構 成ファイル	なし
include.conf	/usr/lpp/rdz/samples/ [/etc/rdz/]	C/C++ コン テンツ・アシ スト用の強制 インクルード	なし
ssl.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	RSE SSL 構 成ファイル	なし
rsecomm.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	RSE トレー ス構成ファイ ル	なし
pushtoclient.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	クライアント 構成ファイル に情報をプッ シュする	なし

IBM Rational Developer for System z Host Utilities、FMID HAKG900

バージョン 8.5 には同等の機能がないため、マイグレーションに関する注意事項はありません。

構成可能なファイル

211 ページの表 27 は、バージョン 9.0 でカスタマイズされるファイルの概要を示しています。Developer for System z Host Utilities のサンプル・ライブラリーの AKG.SAKGSAMP および /usr/lpp/rdzutil/samples には、ここに示したものより多くのカスタマイズ可能なメンバー (サンプルのコード・レビューの後処理スクリプトなど) が含まれます。

注: サンプル・ジョブ AKGSETUP は、リストされているすべてのメンバーを別のデータ・セット (デフォルトでは AKG.#CUST.*) にコピーします。

表 27. Host Utilities バージョン 9.0 のカスタマイズ

メンバーまたはファイル	デフォルト・ロケーション	目的	マイグレーションの注
AKGSETUP	AKG.SAKGSAMP [AKG.#CUST.JCL]	データ・セットを作成し、カスタマイズ可能ファイルのデータを取り込むための JCL	なし
AKGCC	AKG.SAKGSAMP [AKG.#CUST.PROCLIB]	コード・カパレッジの JCL	なし
AKGCR	AKG.SAKGSAMP [AKG.#CUST.PROCLIB]	コード・レビューの JCL	新規 DD BIMPORT および BEXPORT
AKGCRADD	AKG.SAKGSAMP [AKG.#CUST.JCL]	サード・パーティーのコードをコード・レビューに追加するための JCL	なし

バージョン 8.5 のマイグレーションに関する注

以下のマイグレーションに関する注意事項は、バージョン 8.5 に固有のものです。これらの注意事項は、IBM Rational Developer for System z バージョン 8.5.0 からバージョン 8.5.1 へのマイグレーションに該当し、バージョン 8.5.0 のマイグレーションに関する既存の注意事項を補足するものです。

記載している変更内容はすべて、バージョン 8.5.1 以降に該当します。

- CARMA: カスタマイズ可能な CA Endeavor® SCM バッチ・アクションに対する新しいサポートを使用するには、CA Endeavor® SCM RAM 用の CRADEF ファイルおよび CRASTRS VSAM ファイルを更新する必要があります。
- CARMA: CRADEF VSAM 作成時に RAM を使用不可にするサポートが追加されました。
- CARMA: CRASRV.properties の相対ファイル参照のサポートが追加されました。
- CARMA: 新しいサンプル・メンバーが追加されました。
 - CRABJOBC: CA Endeavor® SCM バッチ・アクション用のデフォルト JOB カード。
- CARMA: 以下のカスタマイズ可能メンバーが変更されました。
 - CRASRV.properties
 - carma.startup.rex
 - CRA\$VCAD
 - CRA\$VDEF
 - CRABATCA
 - CRABCFG
 - CRANDVRA
- CARMA: crastart.endeavor.conf および CRASUBCA の CA Endeavor® SCM RAM に対して DD ステートメントがさらに追加されました。

- CRABJCLO (CRANDVRA により割り振られる)
- ENHCEDIT (CRANDVRA により割り振られる)
- カスタマイズ: FEKSETUP JCL が新規メンバーを処理するようになりました。
 - CRABJOBC: FEK.#CUST.CNTL(CRABJOBC) にコピーされました。
- JES ジョブ・モニター - JMON 開始タスクに対して新しいオペレーター・コマンドが追加されました (バージョン 8.0.3.2 より)。
 - MODIFY USERS
 - MODIFY -T{N | E | I | V}
 - MODIFY -M{N | E | W | I | V}
- 問題判別: FEKLOGS JCL は、ユーザー・ログの収集において、複数のユーザー ID の指定をサポートするようになりました。
- 問題判別: FEKLOGS JCL は、問題判別を迅速化するために DD REFORMAT を使用して再フォーマットされたログを収集するようになりました。
- 問題判別: 以下のカスタマイズ可能メンバーが変更されました。
 - FEKLOGS
- RSE: 以下のカスタマイズ不可の新しいディレクティブが rsed.envvars に追加されました。
 - _CMDSERV_BASE_HOME
 - _CMDSERV_CONF_HOME
 - _CMDSERV_WORK_HOME
- RSE: 以下の新しいオプション・ディレクティブが rsed.envvars に追加されました。
 - (_RSE_JAVAOPTS) -DRSE_DSICALL
 - (_RSE_JAVAOPTS) -DDISABLE_REMOTE_INDEX_SEARCH
- RSE: rsed.envvars の以下のオプション・ディレクティブのデフォルト値が変更されました。
 - (_RSE_JAVAOPTS) -Xms
 - (_RSE_JAVAOPTS) -Xmx
 - (_RSE_JAVAOPTS) -Dmaximum.clients
 - (_RSE_JAVAOPTS) -Dmaximum.threads
- RSE: rsed.envvars の以下のカスタマイズ不可のディレクティブのデフォルト値が変更されました。
 - (_RSE_JAVAOPTS) -DDSTORE_SPIRIT_ON
- セキュリティー: 以下の新しいセキュリティー・プロファイルのサポートが追加されました。
 - FEK.USR.**

バージョン 8.0.1 からバージョン 8.5 へのマイグレーション

これらの注は、基本バージョン 8.0.1 からバージョン 8.5 へのマイグレーションのためのものです。これには、バージョン 8.0.1 のメンテナンスの一部として既に文書化されている変更点も含まれています。メンテナンス・ストリームの一部である、したがって既に実装されている可能性がある、変更内容には、それが導入された時点のリリースでマークが付いています。

IBM Rational Developer for System z、FMID HHOP850

- SMP/E による MVS および z/OS UNIX コンポーネントのデフォルトのインストール・ロケーションは、変更されておらず、したがって FEK.* および /usr/lpp/rdz/* のままです。
- CARMA - LPA 内にある CRASTART ロード・モジュールは更新されており、LPA の更新が必要です (バージョン 8.0.3.2 より)。
- CARMA - CRAMSG VSAM を更新する必要があります (バージョン 8.0.3 および 8.5 より)。
- CARMA - CA Endeavor® SCM バッチ・アクション (バージョン 8.0.3 より) および CA Endeavor® SCM パッケージ (バージョン 8.0.3 より) に対する新しいサポートを使用するには、CA Endeavor® SCM RAM 用の CRADEF ファイルおよび CRASTRS VSAM ファイルを更新する必要があります。
- CARMA - 新しい CRADEF および CRASTRS VSAM 入力が追加されており、CA Endeavor® SCM パッケージ・アクションを CA Endeavor® SCM エlement・メニューから復元できるようになっています。
 - CRA0VPKD - CRADEF にマージされます。
 - CRA0VPKS - CRASTRS にマージされます。
- CARMA - 新しいサンプル・メンバーが追加されました (バージョン 8.0.3 より)。
 - CRABCFG - CA Endeavor® SCM バッチ・アクション用の構成ファイル。
 - CRABATCA - CA Endeavor® SCM バッチ・アクション用のサンプル・ジョブ。
- CARMA - 以下のカスタマイズ可能メンバーが変更されました (バージョン 8.0.3、8.0.3.1、および 8.5 より)。
 - CRANDVRA
 - CRASHOW
 - CRASRV.properties
 - CRABCFG
- CARMA - crastart.endeavor.conf および CRASUBCA の CA Endeavor® SCM RAM に対して DD ステートメントがさらに追加されました (バージョン 8.0.3 より)。
 - CRABCFG
 - CRABSKEL
 - PKGSCLS (CRANDVRA により割り振り)
- エンタープライズ・サービス・ツール - IRZ ロード・モジュールおよびメッセージが新しいライブラリーに移動されています (バージョン 8.5 より)。
 - FEK.SFEKLMOD(IRZ* IIRZ*)

- File Manager Integration は削除されました (バージョン 8.5 より)。不定形式 QSAM 編集をはじめとする一部の機能は、Developer for System z による標準データ・セット処理に組み込まれました。コピーブックまたはインクルード・ファイルを使用する定様式データ編集など、さらに上級の機能には、IBM File Manager Plug-in for Eclipse が必要です。
- インクルード・プリプロセッサ - 新しいサンプル・メンバーが追加されました (バージョン 8.0.3.1 より)。
 - FEKRNPLI
- ホスト構成ユーティリティー - マイグレーション・オプションが追加されました (バージョン 8.0.2 より)。
- JES ジョブ・モニター - JMON 開始タスクに対して新しいオペレーター・コマンドが追加されました (バージョン 8.0.3.2 より)。
 - MODIFY STORAGE
- JES ジョブ・モニター - 新しいオプションのディレクティブが FEJCNFG に追加されました (バージョン 8.0.3.1 および 8.0.3.2 より)。
 - LIMIT_CONSOLE
 - SEARCHALL
 - TRACE_STORAGE
- PROCLIB - 以下の PROCLIB メンバーが変更されました (バージョン 8.0.3 より)。
 - ELAXFUOP
- RSE - RSED および LOCKD 開始タスクの始動引数として TMPDIR を指定するオプションが削除されました。このオプションは、/tmp を書き込みアクションで使用できない場合に、開始タスクのユーザー ID のホーム・ディレクトリーを TMPDIR に定義するカスタマイズ不能な機能に置き換えられています (バージョン 8.0.3.1 より)。
- RSE - LOCKD 開始タスクに対して新しいオペレーター・コマンドが追加されました (バージョン 8.0.2 より)。
 - MODIFY DISPLAY TABLE
- RSE - RSED 開始タスクに対して新しいオペレーター・コマンドが追加されました (バージョン 8.0.2、8.0.3、および 8.0.3.2 より)。
 - MODIFY IVP ISPF,userid
 - MODIFY IVP PASSTICKET,userid
 - MODIFY DEBUG HEAPDUMP,PID=pid
 - MODIFY DEBUG JAVACORE,PID=pid
- RSE - RSED 開始タスクのオペレーター・コマンドが拡張されました (バージョン 8.0.2 および 8.0.3.1 より)。
 - MODIFY DISPLAY CLIENT [{,LOGON | ,ID | ,USER}]
 - MODIFY DISPLAY PROCESS,CPU [,PID=pid]
- RSE - 以下のコンソール・メッセージが新たに追加されました (バージョン 8.0.3 および 8.0.3.1 より)。
 - FEK910I = {0} IVP 出口コード = {1} ({0} IVP Exit code = {1})
 - FEK211W ユーザー {0} はログオンしていません。 (User, {0}, not logged on)

- RSE - 新しいカスタマイズ不可能なディレクティブが `rsed.envvars` に追加されました (バージョン 8.0.3 より)。
 - `(_RSE_JAVAOPTS) -Dldap.server.address`
 - `(_RSE_JAVAOPTS) -Dldap.server.port`
 - `(_RSE_JAVAOPTS) -Dldap.ptc.group.name.suffix`
 - `_RSE_PTC`
- RSE - `rsed.envvars` に新しいオプションのディレクティブが追加されました (バージョン 8.0.3、8.0.3.1、および 8.5 より)。
 - `(_RSE_JAVAOPTS) -Daudit.action`
 - `(_RSE_JAVAOPTS) -Daudit.action.id`
 - `(_RSE_JAVAOPTS) -Dlogon.action`
 - `(_RSE_JAVAOPTS) -Dlogon.action.id`
 - `(_RSE_JAVAOPTS) -Dreject.logon.threshold`
 - `(_RSE_JAVAOPTS) -Dinclude.c`
 - `(_RSE_JAVAOPTS) -Dinclude.cpp`
 - `(_RSE_JAVAOPTS) -DCPP_CLEANUP_INTERVAL`
 - `(_RSE_JAVAOPTS) -DRIS_BUFFER`
 - `(_RSE_JAVAOPTS) -DDSTORE_TCP_NO_DELAY`
 - `_RSE_FEK_SAF_CLASS`
 - `_RSE_LDAP_SERVER`
 - `_RSE_LDAP_PORT`
 - `_RSE_LDAP_PTC_GROUP_SUFFIX`
 - `CGI_ISPPREF`
- RSE - 既存の必須ディレクティブが名前変更されました (バージョン 8.5 より)。
 - `_CMDSERV_BASE_HOME` -> `CGI_ISPHOME`
 - `_CMDSERV_CONF_HOME` -> `CGI_ISPCONF`
 - `_CMDSERV_WORK_HOME` -> `CGI_ISPWORK`
 - `_RSE_CMDSERV_OPTS` -> `_RSE_ISPF_OPTS`
- RSE - 既存のオプションのディレクティブが拡張され、さらに多くの値が追加されています (バージョン 8.5 より)。
 - `STEPLIB`
- RSE - `rsed.envvars` 内で以下のオプションのディレクティブの解釈が変更されました (バージョン 8.0.3 より)。
 - `(_RSE_JAVAOPTS) -Dprocess.cleanup.interval`
- RSE - 以下のオプションの構成ファイルが新たに追加されました (バージョン 8.5 より)。
 - `include.conf`
- RSE - 新しいオプションのディレクティブが `pushtoclient.properties` に追加されました (バージョン 8.0.3 より)。
 - `accept.product.license`

- RSE - pushtoclient.properties 内で以下のオプションのディレクティブの解釈が変更されました (バージョン 8.0.3 より)。
 - config.enabled
 - product.enabled
 - reject.config.updates
 - reject.product.updates
- RSE - 新しい z/OS UNIX のサンプルが追加されました (バージョン 8.0.3 および 8.0.3.1 より)。
 - process_audit.rex
 - process_logon.sh
- セキュリティー - 以下の新しいセキュリティー・プロファイルのサポートが追加されました (バージョン 8.0.3 より)。
 - FEK.PTC.**
- zUnit - 以下のオプションの PROCLIB メンバーが新たに追加されました (バージョン 8.5 より)。
 - AZUZUNIT
- 新しい資料、「*IBM Rational Developer for System z メッセージとコード・ガイド*」(SA88-4565)。
- 新しい資料、「*IBM Rational Developer for System z 共通ホスト構成と保守に関する問題への回答*」(SA88-5113)。

構成可能なファイル

217 ページの表 28 は、バージョン 8.5 でカスタマイズされるファイルの概要を示しています。Developer for System z のサンプル・ライブラリー FEK.SFEKSAMP、FEK.SFEKSAMV、および /usr/lpp/rdz/samples/ には、ここに示したものより多くのカスタマイズ可能なメンバー (サンプルの CARMA ソース・コードおよびそれらをコンパイルするジョブなど) が含まれています。

以下のメンバーおよびファイルはもうカスタマイズ可能でないか、使用されなくなっています。

- FMIEXT.properties は使用されなくなりました。

注: サンプル・ジョブ FEKSETUP は、リストされているすべてのメンバーを別のデータ・セットおよびディレクトリー (デフォルトでは FEK.#CUST.* および /etc/rdz/*) にコピーします。

表 28. パージョン 8.5 のカスタマイズ

メンバー/ファイル	デフォルト・ロケーション	目的	マイグレーションの注
FEKSETUP	FEK.SFEKSAMP [FEK.#CUST.JCL]	データ・セットおよびディレクトリーを作成し、カスタマイズ可能ファイルのデータを取り込むための JCL	新しいカスタマイズ可能なメンバーを追加して、新しいディレクトリー構造を作成し、使用されなくなったファイルに対するアクションを除去するために更新された
JMON	FEK.SFEKSAMP (FEJJJCL) [FEK.#CUST.PROCLIB]	JES ジョブ・モニター用の JCL	なし
FEJJJCL	FEK.SFEKSAMP [FEK.#CUST.PROCLIB(JMON)]	JMON メンバーの名前	JMON メンバーを参照
RSED	FEK.SFEKSAMP (FEKRSED) [FEK.#CUST.PROCLIB]	RSE デーモンの JCL	TMPDIR サポートが変更された
FEKRSED	FEK.SFEKSAMP [FEK.#CUST.PROCLIB(RSED)]	RSED メンバーの名前	RSED メンバーを参照
LOCKD	FEK.SFEKSAMP (FEKLOCKD) [FEK.#CUST.PROCLIB]	ロック・デーモン用の JCL	TMPDIR サポートが変更された
FEKLOCKD	FEK.SFEKSAMP [FEK.#CUST.PROCLIB(LOCKD)]	LOCKD メンバーの名前	LOCKD メンバーを参照
ELAXF*	FEK.SFEKSAMP [FEK.#CUST.PROCLIB]	リモート・プロジェクト・ビルドなどのための JCL	メンバー ELAXFUOP が変更された
FEKRACF	FEK.SFEKSAMP [FEK.#CUST.JCL]	セキュリティ定義の JCL	なし
FEJJCNFG	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	JES ジョブ・モニター構成ファイル	オプションの新規ディレクティブが追加された
FEJTSO	FEK.SFEKSAMP [FEK.#CUST.CNTL]	TSO 実行依頼用の JCL	なし
CRA\$VMSG	FEK.SFEKSAMP [FEK.#CUST.JCL]	CARMA メッセージ VSAM を作成するための JCL	VSAM 入力に変更された
CRA\$VDEF	FEK.SFEKSAMP [FEK.#CUST.JCL]	CARMA 構成 VSAM を作成するための JCL	なし

表 28. パージョン 8.5 のカスタマイズ (続き)

メンバー/ファイル	デフォルト・ロケーション	目的	マイグレーションの注
CRA\$VSTR	FEK.SFEKSAMP [FEK.#CUST.JCL]	CARMA カスタム情報 VSAM を作成するための JCL	なし
CRA\$VCAD	FEK.SFEKSAMP [FEK.#CUST.JCL]	CA Endeavor® SCM RAM 用に CARMA 構成 VSAM を作成するための JCL	VSAM 入力に変更 された
CRA\$VCAS	FEK.SFEKSAMP [FEK.#CUST.JCL]	CA Endeavor® SCM RAM 用に CARMA カスタム情 報 VSAM を作成するた めの JCL	VSAM 入力に変更 された
CRASUBMT	FEK.SFEKSAMP [FEK.#CUST.CNTL]	CARMA バッチ始動 CLIST	なし
CRASUBCA	FEK.SFEKSAMP [FEK.#CUST.CNTL]	CA Endeavor® SCM RAM 用の CARMA バッチ始動 CLIST	その他の DD ステ ートメントが追加 された
CRABCFG	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	CA Endeavor® SCM RAM 用の CARMA バッチ・ア クション構成	新規、カスタマイ ズはオプション
CRABATCA	FEK.SFEKSAMP [FEK.#CUST.CNTL]	CA Endeavor® SCM RAM 用の CARMA バッチ・ア クションの JCL	新規、カスタマイ ズはオプション
CRASHOW	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	CA Endeavor® SCM RAM 用の CARMA 構成	新しいフィルター が追加された
CRATMAP	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	CA Endeavor® SCM RAM 用の CARMA 構成	なし
CRANDVRA	FEK.SFEKPROC	CA Endeavor® SCM RAM 用の CARMA 割り振り REXX	その他の DD ステ ートメントが追加 された
CRA#VSLM	FEK.SFEKSAMP [FEK.#CUST.JCL]	SCLM RAM のメッセージ VSAM を作成するための JCL	なし
CRA#ASLM	FEK.SFEKSAMP [FEK.#CUST.JCL]	SCLM RAM のデータ・セ ットを作成するための JCL	なし
CRA#VPDS	FEK.SFEKSAMP [FEK.#CUST.JCL]	PDS RAM のメッセージ VSAM を作成するための JCL	なし
CRA#UADD	FEK.SFEKSAMP [FEK.#CUST.JCL]	RAM 定義をマージするた めの JCL	なし
CRA#UQRY	FEK.SFEKSAMP [FEK.#CUST.JCL]	RAM 定義を抽出するた めの JCL	なし

表 28. パージョン 8.5 のカスタマイズ (続き)

メンバー/ファイル	デフォルト・ロケーション	目的	マイグレーションの注
CRAXJCL	FEK.SFEKSAMP [FEK.#CUST.ASM]	IRXJCL 置換用のサンプル・ソース・コード	なし
CRA#CIRX	FEK.SFEKSAMP [FEK.#CUST.JCL]	CRAXJCL をコンパイルするための JCL	なし
ADNCSDRS	FEK.SFEKSAMP [FEK.#CUST.JCL]	RESTful CRD サーバーを主 CICS 領域に対して定義するための JCL	なし
ADNCSDTX	FEK.SFEKSAMP [FEK.#CUST.JCL]	代替トランザクション ID を CICS 領域に対して定義するための JCL	なし
ADNTXNC	FEK.SFEKSAMP [FEK.#CUST.JCL]	代替トランザクション ID を作成するための JCL	なし
ADNMSGHC	FEK.SFEKSAMP [FEK.#CUST.JCL]	ADNMSGHS をコンパイルするための JCL	なし
ADNMSGHS	FEK.SFEKSAMP [FEK.#CUST.COBOLE]	パイプライン・メッセージ・ハンドラー用のサンプル・ソース・コード	なし
ADNVCRD	FEK.SFEKSAMP [FEK.#CUST.JCL]	CRD リポジトリを作成するための JCL	なし
ADNCSDWS	FEK.SFEKSAMP [FEK.#CUST.JCL]	Web サービス CRD サーバーを主 CICS 領域に対して定義するための JCL	なし
ADNCSDAR	FEK.SFEKSAMP [FEK.#CUST.JCL]	CRD サーバーを非主 CICS 領域に対して定義するための JCL	なし
ADNJSPAU	FEK.SFEKSAMP [FEK.#CUST.JCL]	CRD のデフォルトを更新するための JCL	なし
ADNVMFST	FEK.SFEKSAMP [FEK.#CUST.JCL]	マニフェスト・リポジトリを作成し、定義するための JCL	なし
ELAXMSAM	FEK.SFEKSAMP [FEK.#CUST.PROCLIB]	WLM アドレス・スペースの JCL プロシージャ	なし
ELAXMJCL	FEK.SFEKSAMP [FEK.#CUST.JCL]	PL/I および COBOL ストアード・プロシージャ・ビルダーを DB2 に対して定義するための JCL	なし
AZUZUNIT	FEK.SFEKSAMP [FEK.#CUST.PROCLIB]	zUnit プロシージャの JCL	新規、カスタマイズはオプション

表 28. パージョン 8.5 のカスタマイズ (続き)

メンバー/ファイル	デフォルト・ロケーション	目的	マイグレーションの注
FEKRNPLI	FEK.SFEKSAMP [FEK.#CUST.CNTL]	プリプロセッサ・フレームワーク内部から PL/I コンパイラを呼び出すための REXX	新規、カスタマイズはオプション
FEKLOGS	FEK.SFEKSAMP [FEK.#CUST.JCL]	ログ・ファイルを収集するための JCL	その他の検査が追加された。古いファイルに加えたカスタマイズはすべて再実行する必要がある。
rsed.envvars	/usr/lpp/rdz/samples/ [/etc/rdz/]	RSE 環境変数	古いコピーをこのコピーに置き換え、カスタマイズを再実行する必要がある。
ISPF.conf	/usr/lpp/rdz/samples/ [/etc/rdz/]	TSO/ISPF クライアント・ゲートウェイ構成ファイル	なし
CRASRV.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	CARMA 構成ファイル	一時ポートのサポートが追加された
crastart.conf	/usr/lpp/rdz/samples/ [/etc/rdz/]	CRASTART を使用するための CARMA 構成ファイル	なし
crastart.endevor.conf	/usr/lpp/rdz/samples/ [/etc/rdz/]	CA Endevor® SCM RAM 用に CRASTART を使用するための CARMA 構成ファイル	その他の DD ステートメントが追加された
include.conf	/usr/lpp/rdz/samples/ [/etc/rdz/]	C/C++ コンテンツ・アシスト用の強制インクルード	新規、カスタマイズはオプション
ssl.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	RSE SSL 構成ファイル	なし
rsecomm.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	RSE トレース構成ファイル	一部のディレクティブがオプションになった
pushtoclient.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	クライアント構成ファイルに情報をプッシュする	さらに多くのディレクティブが追加され、既存のディレクティブが拡張された

第 11 章 オペレーター・コマンド

この章では、Developer for System z で使用可能なオペレーター（またはコンソール）コマンドの概要を示します。コマンド・フォーマットの説明に使用される構文図がよく分からない場合、『構文図の読み方』を参照してください。

Start (S)

START コマンドは、開始タスク (STC) を動的に開始するために使用します。このコマンドの省略形バージョンは英字の **S** です。

統合デバッガー

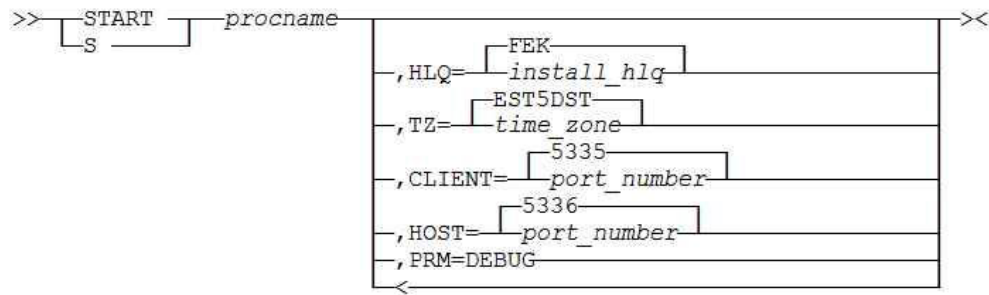


図 39. START DBGMGR オペレーター・コマンド

procname

サーバーを始動するために使用する、プロシージャ・ライブラリー内のメンバーの名前。ホスト・システム構成時に使用されるデフォルト名は DBGMGR です。

HLQ=install_hlq

Developer for System z をインストールするために使用する高位修飾子。デフォルトは FEK です。

TZ=time_zone

タイム・ゾーン・オフセット。デフォルトは EST5DST です。

CLIENT=port_number

外部 (クライアント/ホスト) 通信に使用されるポート番号 (デフォルトは 5335)。

HOST=port_number

内部 (ホスト限定) 通信に使用されるポート (デフォルトは 5336)。

PRM=DEBUG

冗長 (トレース) モードを使用可能にします。トレースはパフォーマンス低下の原因になるため、IBM サポートの指示の下でのみ実行してください。

JES ジョブ・モニター

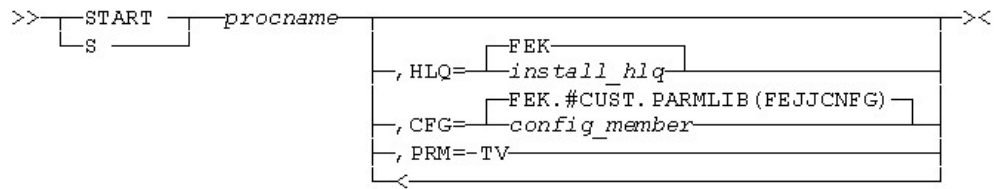


図 40. START JMON オペレーター・コマンド

procname

サーバーを始動するために使用する、プロシージャ・ライブラリー内のメンバーの名前。ホスト・システム構成時に使用されるデフォルト名は JMON です。

HLQ=install_hlq

Developer for System z をインストールするために使用する高位修飾子。デフォルトは FEK です。

CFG=config_member

JES ジョブ・モニター構成ファイルの絶対データ・セットおよびメンバー名。デフォルトは FEK.#CUST.PARMLIB (FEJJCNFG) です。この変数が NULLFILE に設定されている場合、JES ジョブ・モニターではデフォルト構成値が使用されます。

PRM=-TV

冗長 (トレース) モードを使用可能にします。トレースはパフォーマンス低下の原因になるため、IBM サポートの指示の下でのみ実行してください。

RSE デーモン

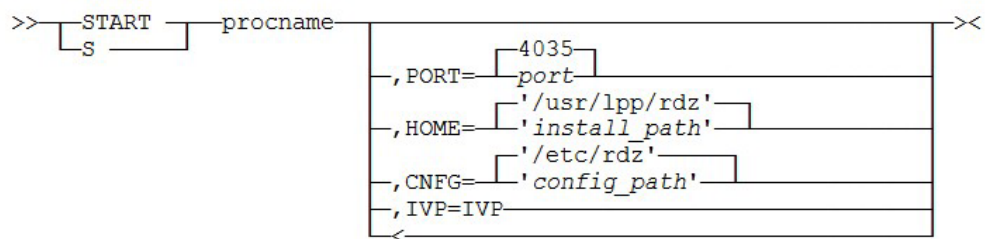


図 41. START RSED オペレーター・コマンド

procname

サーバーを始動するために使用する、プロシージャ・ライブラリー内のメンバーの名前。ホスト・システム構成時に使用されるデフォルト名は RSED です。

PORT=port

RSE デーモンがクライアントの接続に使用するポート。指定しない場合、変数 _RSE_RSED_PORT の /etc/rdz/rsed.envvars で定義されたポートが使用されます。デフォルトは 4035 です。

IVP=IVP

サーバーを始動しませんが、RSE デーモン・インストール検査プログラム (IVP) を実行します。

CNFG='config_path'

z/OS UNIX 内に保管される構成ファイルの絶対ロケーション。デフォルトは '/etc/rdz' です。z/OS UNIX パスには大/小文字の区別があり、小文字を保持する場合は単一引用符 (') で囲む必要があります。

HOME='install_path'

Developer for System z をインストールするために使用するパス接頭部および必須の /usr/lpp/rdz。デフォルトは '/usr/lpp/rdz' です。z/OS UNIX パスには大/小文字の区別があり、小文字を保持する場合は単一引用符 (') で囲む必要があります。

Modify (F)

MODIFY コマンドを使用して、アクティブ・タスクの特性を動的に照会および変更できます。このコマンドの省略形バージョンは英字の F です。

統合デバッガー

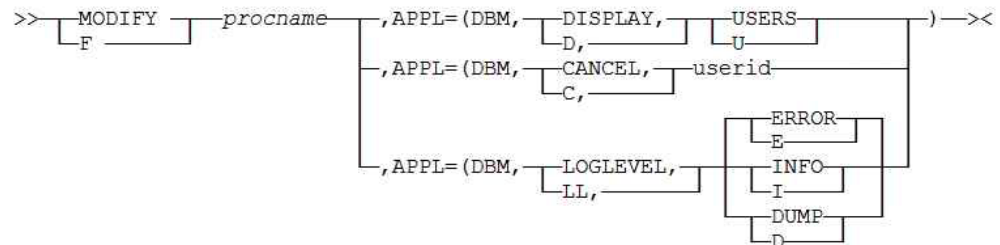


図 42. MODIFY DBGMGR オペレーター・コマンド

procname

サーバーを始動するために使用する、プロシージャ・ライブラリー内のメンバーの名前。ホスト・システム構成時に使用されるデフォルト名は DBGMGR です。

DISPLAY,USERS

単一の複数行 AQECM104I コンソール・メッセージを使用してアクティブ・ユーザーを表示します。アクティブ・ユーザーが存在しない場合はメッセージ AQECM103I が発行されます。ユーザー・リストは、サーバーにおけるそのユーザーの状態を表示します。統合デバッガーのデータ・フローの概要については、「ホスト構成リファレンス」(SA88-4226-03) の

『Developer for System z について』の章の『統合デバッガー』のセクションを参照してください。

```
AQECM104I
User:IBMUSER RegisterSocket(2)
User:IBMUSR2 18354752 ProbeSocket(3) waits for register connection
```

```
User:IBMUSR3 25387329 ProbeSocket(5) waits for engine connection
User:IBMUSR4 24113603 Engine(4) connected to Probe(8)
Module(AQETST)
AQECM103I There is no active user
```

最初のメッセージ (IBMUSER 用) は、ユーザーが登録済みであり、デバッグ・アクティビティーが存在しないことを示しています。2 番目のメッセージ (IBMUSR2 用) は、デバッグ・セッションがユーザーの登録を待機中であることを示しています。3 番目のメッセージ (IBMUSR3 用) は、デバッグ・セッションがセットアップされていることを示しています。4 番目のメッセージ (IBMUSR4 用) は、モジュール AQETST 用のアクティブなデバッグ・セッションを示しています。

CANCEL,userid

指定されたユーザー ID に関するすべてのデバッグ・セッションをキャンセルします。結果は AQECM110I または AQECM111I コンソール・メッセージで表示されます。

```
AQECM110I user(IBMUSER) canceled
AQECM111I user(IBMUSER) not connected
```

LOGLEVEL,{ERROR | INFO | DUMP}

デバッグ・マネージャーのメッセージ・ログ (DD SYSPRINT) の詳細レベルを制御します。デフォルトは E (エラー) です。メッセージ「LOGLEVEL コマンドが正常に処理されました」が、メッセージ ID AQECM101I でコンソールに書き込まれます。

E または ERROR	エラー・メッセージのみ (デフォルト)
I または INFO	エラー・メッセージおよび通知メッセージ
D または DUMP	エラー・メッセージ、通知メッセージ、およびデバッグ/ダンプ・メッセージ

詳細なトレースはパフォーマンス低下の原因になるため、IBM サポートの指示の下でのみ実行してください。

JES ジョブ・モニター

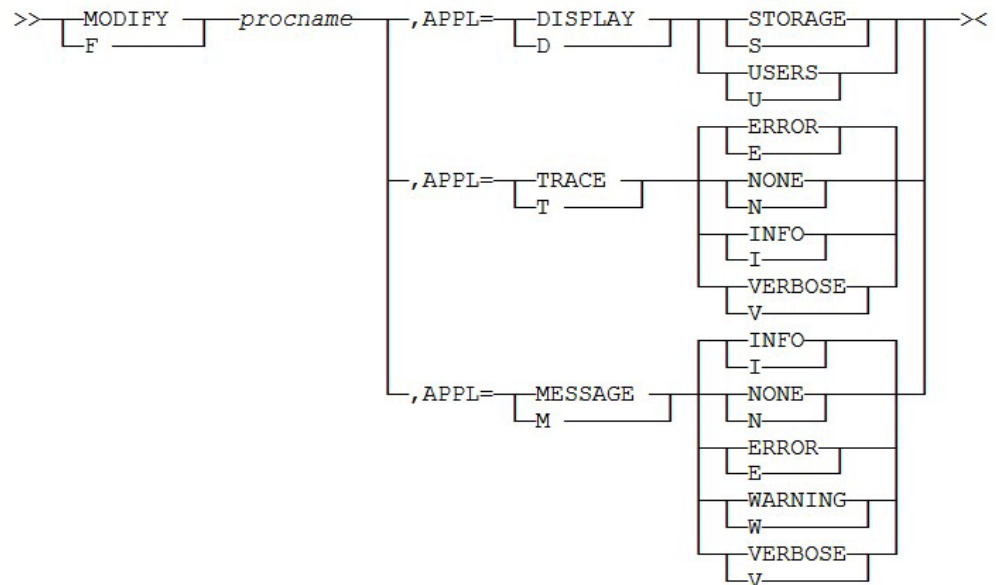


図 43. MODIFY JMON オペレーター・コマンド

procname

サーバーを始動するために使用された、プロシージャ・ライブラリー内のメンバーの名前。ホスト・システム構成時に使用されるデフォルト名は JMON です。

DISPLAY STORAGE

ストレージ使用量レポートを DD SYSOUT に書き込みます。メッセージ「JMON のストレージ情報が SYSOUT に書き込まれました (JMON storage information written to SYSOUT)」が、メッセージ ID BPXM023I でコンソールに書き込まれます。ストレージ使用量レポートには、バイト単位、キロバイト単位、およびメガバイト単位のサイズを示すさまざまなストレージ関連フィールドが表示されます。

```

>>>STORAGE TRACE (console request)<<<
LDAREGRQ 00000000000 00000000K 00000M requested region size
         below 16M line
LDASIZA 00006266880 00006120K 00005M maximum region size
LDALIMIT 00006266880 00006120K 00005M limit
LDAVVRG 00006266880 00006120K 00005M getmain limit
LDALOAL 00000061440 00000060K 00000M in use
LDAHIAL 00000266240 00000260K 00000M LSQA/SWA/private subpools
 _GAP 00000000000 00000000K 00000M gaps in allocation
 _AVAIL 00005939200 00005800K 00005M available (including gaps)
 _MAX 00006000640 00005860K 00005M current limit
         above 16M line
LDAESIZA 01905262592 01860608K 01817M maximum region size
LDAELIM 01905262592 01860608K 01817M limit
LDAEVVRG 01905262592 01860608K 01817M getmain limit
LDAELOAL 00000937984 00000916K 00000M in use
LDAEHIAL 00012754944 00012456K 00012M ELSQA/ESWA/private subpools
 _EGAP 00000000000 00000000K 00000M gaps in allocation
 _EAVAIL 01891569664 01847236K 01803M available (including gaps)
 _EMAX 01892507648 01848152K 01804M current limit
  
```

DISPLAY USERS

アクティブ・ユーザーのリストを DD SYSOUT に書き込みます。メッセージ「JMON のユーザー・リストが SYSOUT に書き込まれました (JMON user list written to SYSOUT)」が、メッセージ ID BPXM023I でコンソールに書き込まれます。ユーザー・リストには、CPU 使用量を含む、さまざまなユーザー関連データが示されます。

```
S0  userid    USER      4:04(elapsed)    4:04(idle)
Users: 1
```

TRACE {NONE | ERROR | INFO | VERBOSE}

JES ジョブ・モニターのトレース・ログ (DD SYSOUT) の詳細レベルを制御します。デフォルトは E (エラー) です。メッセージ「JMON トレース・レベル:{なし | エラー | 情報 | 詳細} (JMON TRACE LEVEL:{NONE | ERROR | INFO | VERBOSE})」が、メッセージ ID BPXM023I でコンソールに書き込まれます。

N または NONE	始動メッセージのみ
E または ERROR	始動メッセージおよびエラー・メッセージのみ (デフォルト)
I または INFO	始動メッセージ、エラー・メッセージ、および情報メッセージ
V または VERBOSE	始動メッセージ、エラー・メッセージ、情報メッセージ、および詳細メッセージ

詳細なトレースはパフォーマンス低下の原因になるため、IBM サポートの指示の下でのみ実行してください。

MESSAGE {NONE | ERROR | WARNING | INFO | VERBOSE}

JES ジョブ・モニターのメッセージ・ログ (DD SYSPRINT) の詳細レベルを制御します。デフォルトは I (情報) です。メッセージ「JMON メッセージ・レベル:{なし | エラー | 警告 | 情報 | 詳細} (JMON MESSAGE LEVEL:{NONE | ERROR | WARNING | INFO | VERBOSE})」が、メッセージ ID BPXM023I でコンソールに書き込まれます。

N または NONE	メッセージはありません。
E または ERROR	エラー・メッセージのみ
W または WARNING	エラー・メッセージおよび警告メッセージ
I または INFO	エラー・メッセージ、警告メッセージ、および情報メッセージ (デフォルト)
V または VERBOSE	エラー・メッセージ、警告メッセージ、情報メッセージ、および詳細メッセージ

詳細なトレースはパフォーマンス低下の原因になるため、IBM サポートの指示の下でのみ実行してください。

RSE デーモン

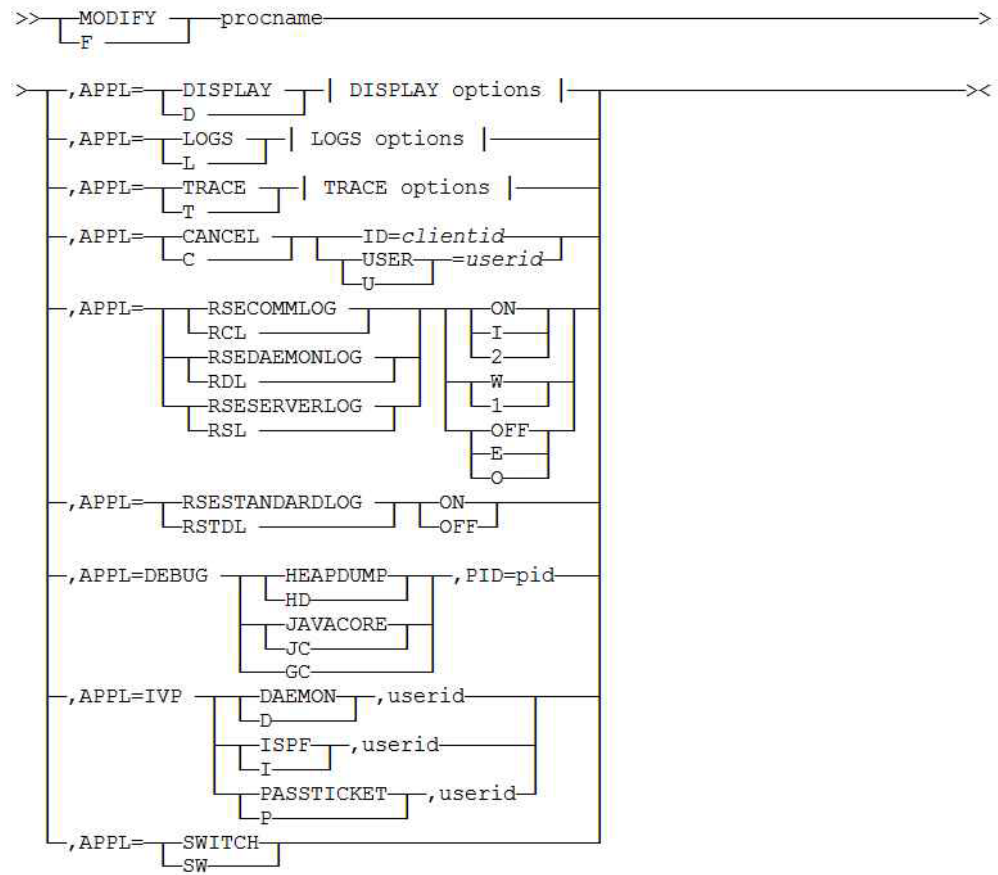


図 44. MODIFY RSED オペレーター・コマンド

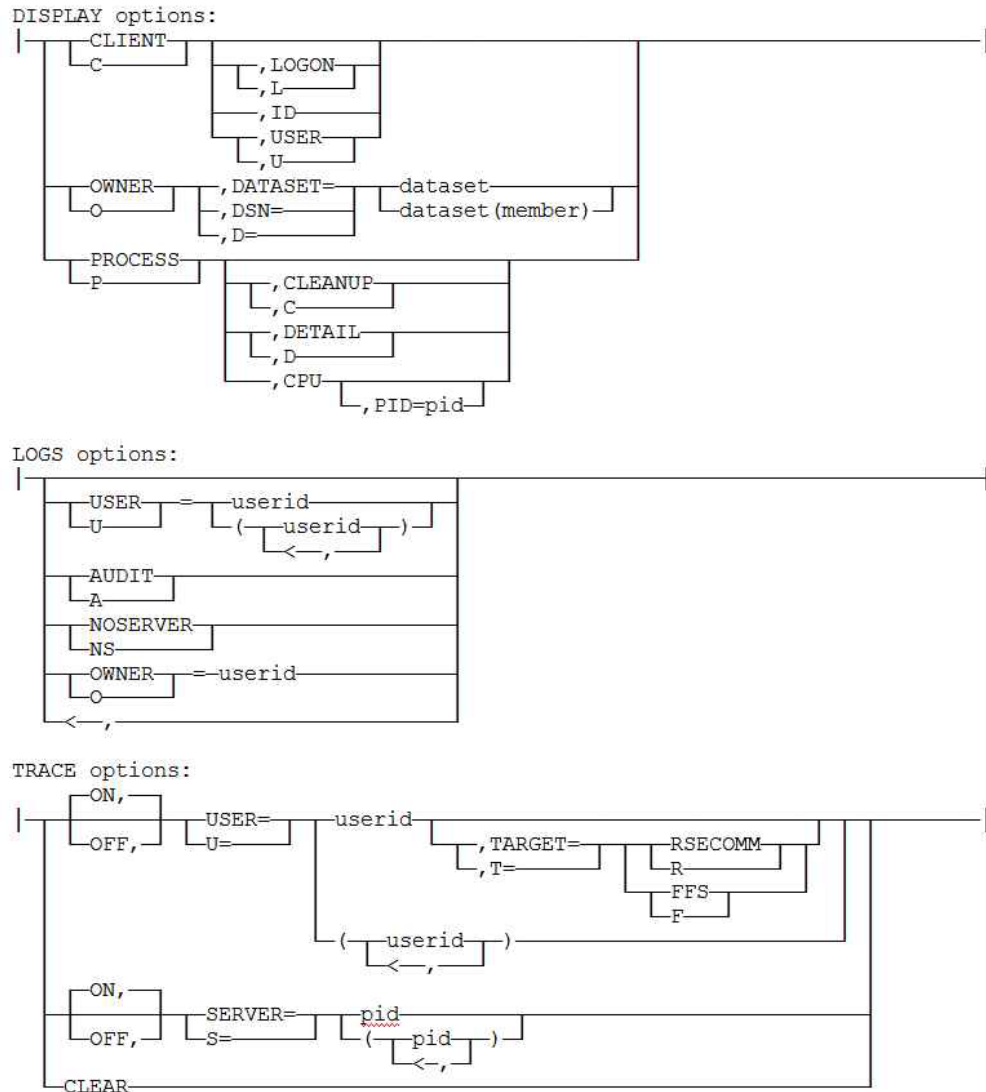


図 45. MODIFY RSED オペレーター・コマンド (続き)

procname

サーバーを始動するために使用された、プロシージャー・ライブラリー内のメンバーの名前。ホスト・システム構成時に使用されるデフォルト名は RSED です。

DISPLAY CLIENT [{,LOGON | ,ID | ,USER}]

単一 BPXM023I メッセージ内のアクティブなクライアントを表示します。結果のレイアウトは、使用されたコマンド・オプションによって異なります。オプションのコマンド引数を使用して、ソート順を変更することができます。

- コマンド・オプションなし: クライアントは、クライアントにサービスを提供するスレッド・プール・プロセスでグループ化されます。

ProcessId(<processid>) ASId(<asid>) JobName(<jobname>)
Clients(<local>/<total>) Order(<startup order>)
<clientid><userid><connected since>

- LOGON コマンド・オプション: クライアントは、ログオン時間で順序付けされます。

```
LOGON TIME----- ID----- USERID--
<connected since>      <clientid> <userid>
```

- ID コマンド・オプション: クライアントは、クライアント ID で順序付けされます。

```
ID----- USERID-- LOGON TIME-----
<clientid> <userid> <connected since>
```

- USER コマンド・オプション: クライアントは、ユーザー ID で順序付けされます。

```
USERID-- ID----- LOGON TIME-----
<userid> <clientid> <connected since>
```

DISPLAY OWNER,DATASET={dataset | dataset(member)}

単一の BPXM023I メッセージに、データ・セットのエンキュー所有者を表示します。

```
FEK217I <dataset[(member)]> is locked by <userid>
FEK218I <dataset[(member)]> is not locked
FEK219E Failed to determine lock owner for <dataset[(member)]>
```

- サーバーからは、ISPF などの他の製品によって保持されているロックも報告されます。
- **D GRS,RES=(*,dataset)** オペレーター・コマンドでは、どの Developer for System z ユーザーが実際のエンキュー所有者であるかを示すことはできません。このコマンドで示すことができるのは、ユーザーがアクティブなスレッド・プールのみです。

DISPLAY PROCESS[{,CLEANUP | ,CPU [,PID=pid] | ,DETAIL}]

1 つ以上の BPXM023I メッセージに RSE スレッド・プール・プロセスを表示します。接続したユーザーのロード・バランシングに複数のプロセスが使用されている可能性があります。

```
ProcessId(<processid>) Memory Usage(<java heap usage>%)
Clients(<number of clients>) Order(<startup order>) <error status>
```

注:

- <processid> は、プロセス固有の z/OS UNIX オペレーター・コマンドで使用できます。
- 各プロセスには独自の Java ヒープがあり、そのサイズは `rsed.envvars` の中で設定できます。報告される Java ヒープの使用量には、Developer for System z から解放されたものの、Java のガーベッジ・コレクション・プロセスからはまだ解放されていないストレージが含まれています。
- <startup order> は、スレッド・プールが開始された順序を示す連番です。この番号は、`stderr.*.log` および `stdout.*.log` ファイルのファイル名に使用されている数字と一致します。

通常の状態では、<error status> はブランクです。230 ページの表 29 に、<error status> のブランク以外の値を示します。

表 29. スレッド・プールのエラー状況

状況	説明
severe error	スレッド・プール・プロセスでリカバリー不能エラーが発生し、操作が停止されました。その他の状況フィールドには、最後に認識された値が示されます。この項目をテーブルから除去するには、 DISPLAY PROCESS 変更コマンドの CLEANUP オプションを使用します。
killed process	スレッド・プール・プロセスが、Java、z/OS UNIX、またはオペレーター・コマンドによって強制終了されました。その他の状況フィールドには、最後に認識された値が示されます。この項目をテーブルから除去するには、 DISPLAY PROCESS 変更コマンドの CLEANUP オプションを使用します。
timeout	クライアント接続要求で、スレッド・プール・プロセスが時間内に RSE デーモンに回答しませんでした。その他の状況フィールドには、現行値が示されます。このスレッド・プールは、今後のクライアント接続要求で除外されます。 *timeout* 状況は、このスレッド・プールで処理されているクライアントがログオフするとリセットされます。
rejectLogon	ワークロードが高いことが原因で、スレッド・プールで一時的にログオン要求を受け入れられなくなります。ログオン要求は他のスレッド・プールに引き渡されます。クライアントへの大きいファイルのアップロードなど、リソースに負荷のかかるタスクが完了すると、 *rejectLogon* 状況はリセットされます。

DISPLAY PROCESS 変更コマンドの **DETAIL** オプションを使用すると、詳細情報が表示されます。

```

ProcessId(33555087) ASId(002E) JobName(RSED8) Order(1)
PROCESS LIMITS:    CURRENT  HIGHWATER    LIMIT
  JAVA HEAP USAGE(%)    10       56       100
    CLIENTS              0        25        30
  MAXFILEPROC           83       103     64000
  MAXPROCUSER           97        99       200
  MAXTHREADS             9        14      1500
  MAXTHREADTASKS         9        14      1500

```

ASId フィールドは、16 進表記のアドレス・スペース ID です。

「PROCESS LIMITS」の表には、現在のリソース使用量、リソース使用量の最高水準点、およびリソースの限度が示されます。他の制限要因のために、定義された限度に到達しない場合があります。

DISPLAY PROCESS 変更コマンドの **CPU** オプションにより、スレッド・プール内のスレッドごとに累積 CPU 使用量がミリ秒単位で示されます。各スレッド・プールに BPXM023I メッセージがあります。デフォルトで、すべて

のスレッド・プールが CPU 使用量を報告しますが、オペレーター・コマンドで PID=pid を指定することによって範囲を単一スレッド・プールに制限することができます。ここで、pid は対象のスレッド・プールのプロセス ID です。

```
ProcessId(421      ) ASId(007D) JobName(RSED8) Order(1)
USERID  THREAD-ID      TCB@      ACC_TIME TAG
STCRSE  0EDE54000000000 005E6B60      822 1/ThreadPoolProcess
STCRSE  0EDE87000000000 005E69C8       001
STCRSE  0EDE98000000000 005E6518      1814
STCRSE  0EDEBA000000000 005E66B0      2305
STCRSE  0EDECB000000000 005E62F8       001
STCRSE  0EDED0000000000 005E60D8       001
STCRSE  0EDF86000000000 005C2BF8      628 6/ThreadPoolMonitor$Memory
UsageMonitor
STCRSE  0EDF97000000000 005C2D90       003 7/ThreadPoolMonitor
STCRSE  0EDFDB000000000 005C29D8       001
STCRSE  0EE22E000000000 005C1BE0       070
IBMUSER 0EE0EB000000000 005C22B8      276 20/ServerReceiver
IBMUSER 0EE250000000000 005C19C0      137 16/ServerUpdateHandler
IBMUSER 0EE261000000000 005C17A0      509 15/ServerCommandHandler
IBMUSER 0EE184000000000 005C1E00      065 21/ZosSystemMiner
STCRSE  0EE151000000000 005C2098       078
STCRSE  0EE195000000000 005C1580       001
IBMUSER 0EE23F000000000 005C1360      021 26/UniversalFileSystemMine
r
IBMUSER 0EE2A5000000000 005C0CF0       003 27/EnvironmentMiner
IBMUSER 0EE283000000000 005C1140       002 31/CommandMiner
IBMUSER 0EE272000000000 005C0E88      081 32/MVSFileSystemMiner
IBMUSER 0EE294000000000 005C0AD0      002 33/MVSByteStreamHandler$Op
enCloseThread
STCRSE  0EE2E9000000000 005C0470       001
IBMUSER 0EE2C7000000000 005C08B0      050 38/JESMiner
IBMUSER 0EE2B6000000000 005C0690      004 40/FAMiner
IBMUSER 0EE30B000000000 005C0250      002 41/LuceneMiner
IBMUSER 0EE31C000000000 005C0030      002 42/CDTParserMiner
IBMUSER 0EE32D000000000 005BDE00      002 43/MVSLuceneMiner
IBMUSER 0EE33E000000000 005BDBE0      002 44/CDTMVSParserMiner
```

出力のサイズがコンソール・メッセージの最大行数を超える場合、出力は複数の BPXM023I メッセージに分割されます。これらの追加のメッセージには最初のメッセージと同じヘッダーが示されますが、1 行目に CONTINUATION キーワードが追加されています。

```
ProcessId(421      ) ASId(007D) JobName(RSED8) Order(1) CONTINUATION
USERID  THREAD-ID      TCB@      ACC_TIME TAG
```

出力は、各スレッド・プールで最初の 4000 スレッドまでに限定されています。

CANCEL ID=clientid

クライアント ID に基づいて、クライアント接続をキャンセルします。クライアント ID は **DISPLAY CLIENT** 変更コマンドで表示されます。

クライアント接続をキャンセルすると、ホスト・システム・スレッドは、正常な終了処理を経て、使用していたリソースをクリーンアップします。このアクションがあるため、スレッドによっては、終了までに数分かかるものがあります (キープアライブ・メカニズムによるタイムアウトを待機しているスレッドなど)。

CANCEL USER=userid

クライアントのユーザー ID に基づいて、クライアント接続をキャンセルします。クライアントのユーザー ID は **DISPLAY CLIENT** 変更コマンドで表示されます。

クライアント接続をキャンセルすると、ホスト・システム・スレッドは、正常な終了処理を経て、使用していたリソースをクリーンアップします。このアクションがあるため、スレッドによっては、終了までに数分かかるものがあります (キープアライブ・メカニズムによるタイムアウトを待機しているスレッドなど)。

LOGS [USER={userid | (userid,userid,...)},]

[AUDIT,][NOSERVER,][OWNER=userid,]

Developer for System z ホスト・ログおよびセットアップ情報を収集します。収集したデータは、z/OS UNIX ファイルである \$TMPDIR/feklogs %sysname.%jobname に配置されます。ここで、\$TMPDIR は rsed.envvars の TMPDIR ディレクティブの値 (デフォルトは /tmp)、%sysname はご使用の z/OS システムの名前、%jobname は RSED 開始タスクの名前です。結果は、単一の FEK201I コンソール・メッセージに表示されます。

FEK220I ホスト・ログは /tmp/feklogs.CDFMVS08.RSED.log
に書き込まれます

(Host logs are written onto /tmp/feklogs.CDFMVS08.RSED.log)

デフォルトでは、サーバー・ログだけが収集されます。コマンド・オプションを使用すると、さまざまなログを収集できます。

USER	指定したユーザー ID のログ・ファイルを収集する
AUDIT	監査ログを収集する
NOSERVER	サーバー・ログを収集しない

Developer for System z は、セキュリティ製品で FEK.CMD.LOGS.** プロファイルに対するアクセス権を照会し、指定されたログの収集をリクエスターが許可されているかどうかを判別します。デフォルトでは、リクエスターは RSED 開始タスクのユーザー ID です (OWNER オプションが指定されている場合を除く)。リクエスターだけが、収集したデータを収容しているファイルに対するアクセス権を保持します。

RSECOMMLOG {ON | OFF | I | W | E | 2 | 1 | 0}

RSE サーバー (rsecomm.log) および MVS データ・セット・サービス (lock.log および ffs*.log) のトレース詳細レベルを制御します。始動デフォルトは rsecomm.properties で定義されます。次の 3 つの詳細レベルが使用可能です。

E または 0 または OFF	エラー・メッセージのみ。
W または 1	エラー・メッセージと警告メッセージ。これは、rsecomm.properties でのデフォルトの設定です。
I または 2 または ON	エラー・メッセージ、警告メッセージ、および情報メッセージを記録します。

詳細なトレースはパフォーマンス低下の原因になるため、IBM サポートの指示の下でのみ実行してください。

RSEDAEMONLOG {ON | OFF | I | W | E | 2 | 1 | 0}

RSE デーモン (rsedaemon.log) のトレース詳細レベルを制御します。始動デフォルトは rsecomm.properties で定義されます。以下の 3 つの詳細レベルを使用できます。

E または 0 または OFF	エラー・メッセージのみ。
W または 1	エラー・メッセージと警告メッセージ。これは、rsecomm.properties でのデフォルトの設定です。
I または 2 または ON	エラー・メッセージ、警告メッセージ、および情報メッセージを記録します。

詳細なトレースはパフォーマンス低下の原因になるため、IBM サポートの指示の下でのみ実行してください。

RSESERVERLOG {ON | OFF | I | W | E | 2 | 1 | 0}

RSE スレッド・プール (rseserver.log) のトレース詳細レベルを制御します。始動デフォルトは rsecomm.properties で定義されます。次の 3 つの詳細レベルが使用可能です。

E または 0 または OFF	エラー・メッセージのみ。
W または 1	エラー・メッセージと警告メッセージ。これは、rsecomm.properties でのデフォルトの設定です。
I または 2 または ON	エラー・メッセージ、警告メッセージ、および情報メッセージを記録します。

詳細なトレースはパフォーマンス低下の原因になるため、IBM サポートの指示の下でのみ実行してください。

RSESTANDARDLOG {ON | OFF}

stdout.*,log スレッド・プールおよび stderr.*,log スレッド・プールの stdout ストリームおよび stderr ストリームを保持しているログ・ファイルの更新を、無効 (OFF) または有効 (ON) にします。始動デフォルトは、rsed.envvars 内の enable.standard.log ディレクティブで定義されます。

詳細なトレースはパフォーマンス低下の原因になるため、IBM サポートの指示の下でのみ実行してください。

TRACE [{ON, | OFF,}]USER=userid[,TARGET={FFS | RSECOMM}]

指定されたユーザー ID に関するトレースを有効 (ON) または無効 (OFF) にします。デフォルトは ON です。この設定は、**MODIFY RSECOMMLOG** オペレーター・コマンドによって制御されるデフォルト設定を却下します。次の 2 つの詳細レベルが使用可能です。

オフ	エラー・メッセージのみ
ON (デフォルト)	エラー・メッセージ、警告メッセージ、および情報メッセージを記録します。

このコマンドは、RSE サーバー (rsecomm.log) および MVS データ・セット・サービス (lock.log と ffs*.log) のトレース詳細レベルを変更します。以下の 2 つの値を受け入れる TARGET キーワードを使用して、これを限定することができます。

FFS	MVS データ・セット・サービス (lock.log と ffs*.log) に対してのみ、指定されたログ・レベルを設定します。
RSECOMM	RSE サーバー (rsecomm.log) に対してのみ、指定されたログ・レベルを設定します。

現在ログオンしていないユーザーに関して、このコマンドを発行することができます。この設定はユーザーがログオフしてもアクティブのままになり、ユーザーがログオンすると再び使用されます。

サーバー始動時に **MODIFY TRACE USER** コマンドの発行をシミュレートするには、rsecomm.properties 内で USER ディレクティブを使用します。オペレーター・コマンド **MODIFY TRACE USER** または **MODIFY TRACE SERVER**、あるいは rsecomm.properties 内の USER ディレクティブによって既に設定されている場合は、このコマンドによる設定で置き換えられます。

詳細なトレースはパフォーマンス低下の原因になるため、IBM サポートの指示の下でのみ実行してください。

TRACE [{ON, | OFF,}]USER=(userid,userid,...)

指定されたユーザー ID に関するトレースを有効 (ON) または無効 (OFF) にします。デフォルトは ON です。この設定は、**MODIFY RSECOMMLOG** オペレーター・コマンドによって制御されるデフォルト設定を却下します。次の 2 つの詳細レベルが使用可能です。

オフ	エラー・メッセージのみ。
ON (デフォルト)	エラー・メッセージ、警告メッセージ、および情報メッセージを記録します。

このコマンドは、RSE サーバー (rsecomm.log) および MVS データ・セット・サービス (lock.log と ffs*.log) のトレース詳細レベルを変更します。現在ログオンしていないユーザーに関して、このコマンドを発行することができます。この設定はユーザーがログオフしてもアクティブのままになり、ユーザーがログオンすると再び使用されます。サーバー始動時に **MODIFY TRACE USER** コマンドの発行をシミュレートするには、rsecomm.properties 内で USER ディレクティブを使用します。オペレーター・コマンド **MODIFY TRACE USER** または **MODIFY TRACE SERVER**、あるいは rsecomm.properties 内の USER ディレクティブによって既に設定されている場合は、このコマンドによる設定で置き換えられます。

詳細なトレースはパフォーマンス低下の原因になるため、IBM サポートの指示の下でのみ実行してください。

TRACE [{ON, | OFF,}] SERVER={pid | (pid,pid,...)}

指定されたスレッド・プール内のすべてのユーザーに関するトレースを有効 (ON) または無効 (OFF) にします。pid は RSE スレッド・プールのプロセス

ID です。デフォルトは ON です。この設定は、**MODIFY RSECOMMLOG** オペレーター・コマンドによって制御されるデフォルト設定を却下します。次の 2 つの詳細レベルが使用可能です。

オフ	エラー・メッセージのみ。
ON (デフォルト)	エラー・メッセージ、警告メッセージ、および情報メッセージを記録します。

このコマンドは、RSE サーバー (rsecomm.log) および MVS データ・セット・サービス (lock.log と ffs*.log) のトレース詳細レベルを変更します。オペレーター・コマンド **MODIFY TRACE USER** または **MODIFY TRACE SERVER**、あるいは rsecomm.properties 内の USER ディレクティブによって既に設定されている場合は、このコマンドによる設定で置き換えられます。

詳細なトレースはパフォーマンス低下の原因になるため、IBM サポートの指示の下でのみ実行してください。

TRACE CLEAR

オペレーター・コマンド **MODIFY TRACE USER** と **MODIFY TRACE SERVER**、および rsecomm.properties 内の USER ディレクティブで設定されたすべてのトレース・オーバーライドを除去します。

DEBUG HEAPDUMP,PID=pid

指定されたスレッド・プールの Java ヒープ・ダンプを要求します。ここで、pid は RSE スレッド・プールのプロセス ID です。ダンプは rsed.envvars の _CEE_DUMPTARG によって指定されたディレクトリーに書き込まれます。デフォルト値は /tmp です。結果は単一の BPXM023I コンソール・メッセージに表示されます。

```
JVMDUMP034I User requested Heap dump using '/tmp/heapdump.20120223.211'
430.16777590.0001.phd' through JVMRI
```

DEBUG JAVACORE,PID=pid

指定されたスレッド・プールの Java コア・ダンプを要求します。ここで、pid は RSE スレッド・プールのプロセス ID です。ダンプは rsed.envvars の _CEE_DUMPTARG によって指定されたディレクトリーに書き込まれます。デフォルト値は /tmp です。結果は単一の BPXM023I コンソール・メッセージに表示されます。

```
JVMDUMP034I User requested Java dump using '/tmp/javacore.20120223.214
244.16777590.0002.phd' through JVMRI
```

DEBUG GC,PID=pid

指定されたスレッド・プールの Java ガーベッジ・コレクションを要求します。ここで、pid は RSE スレッド・プールのプロセス ID です。

IVP DAEMON,userid

接続テストを行うために、ユーザー ID userid を RSE デーモンにログオンさせます。結果は 1 つ以上の FEK900I コンソール・メッセージとともに表示されます。戻りコードは、コンソール・メッセージ FEK901I とともに表示されます。

```
+FEK900I DAEMON IVP: SSL is disabled
+FEK900I DAEMON IVP: connected
+FEK900I DAEMON IVP: 1977
+FEK900I DAEMON IVP: 6902918
+FEK900I DAEMON IVP: Success
+FEK901I DAEMON IVP Exit code = 0
```

注:

- この機能は、fekfivpd IVP (インストール検査プログラム) の機能に似ています。
- RSE デーモンにより、IVP のパスワードとして使用する PassTicket が生成されるため、パスワードを要求する要応答オペレーター宛メッセージ (WTOR) はありません。

IVP ISPF,userid

ISPF のクライアント・ゲートウェイをユーザー ID userid として呼び出します。結果は 1 つ以上の FEK900I コンソール・メッセージとともに表示されます。戻りコードは、コンソール・メッセージ FEK901I とともに表示されます。

```
+FEK900I ISPF IVP: executed on CDFMVS08 -- Tue Sep 13 22:29:28 EDT 2011
+FEK900I ISPF IVP: executed by uid=1(IBMUSER) gid=0(SYS1)
+FEK900I ISPF IVP: using /etc/rdz/rsed.envvars
+FEK900I ISPF IVP: current address space size limit is 2147483647
(2048.0 MB)
+FEK900I ISPF IVP: maximum address space size limit is 2147483647
(2048.0 MB)
+FEK900I ISPF IVP: -----
-----
+FEK900I ISPF IVP: /etc/rdz/ISPF.conf content:
+FEK900I ISPF IVP: -----
-----
+FEK900I ISPF IVP: ispllib=ISP.SISPLOAD
+FEK900I ISPF IVP: ispmllib=ISP.SISPMENU
+FEK900I ISPF IVP: isptlib=ISP.SISPTENU
+FEK900I ISPF IVP: ispplib=ISP.SISPPENU
+FEK900I ISPF IVP: ispslib=ISP.SISPSLIB
+FEK900I ISPF IVP: sysproc=ISP.SISPCLIB,FEK.SFEKPROC
+FEK900I ISPF IVP: -----
-----
+FEK900I ISPF IVP: Host install verification for RSE
+FEK900I ISPF IVP: Review IVP log messages from HOST below :
+FEK900I ISPF IVP: -----
-----
+FEK900I ISPF IVP: Service level 22Feb2011
+FEK900I ISPF IVP: RSE connection and base TSO/ISPF session initialization
on check only
+FEK900I ISPF IVP: *** CHECK : ENVIRONMENT VARIABLES - key variables
displayed below :
+FEK900I ISPF IVP: Server PATH = ./usr/lpp/java/J6.0/bin:/usr/l
pp/rdz/bin:/usr/lpp/ispf/bin:/bin:/usr/sbin
+FEK900I ISPF IVP: STEPLIB = NONE
+FEK900I ISPF IVP: Temporary directory = /tmp
+FEK900I ISPF IVP: CGI_ISPHOME = /usr/lpp/ispf
+FEK900I ISPF IVP: CGI_ISPCONF = /etc/rdz
+FEK900I ISPF IVP: CGI_ISPWORK = /var/rdz
+FEK900I ISPF IVP: -----
-----
+FEK900I ISPF IVP: *** CHECK : USS MODULES
+FEK900I ISPF IVP: Checking ISPF Directory : /usr/lpp/ispf
+FEK900I ISPF IVP: Checking modules in /usr/lpp/ispf/bin directory
+FEK900I ISPF IVP: Checking for ISPF configuration file ISPF.conf
+FEK900I ISPF IVP: RC=0
```

```

+FEK900I ISPF IVP: MSG: SUCCESSFUL
+FEK900I ISPF IVP: -----
-----
+FEK900I ISPF IVP: *** CHECK : TSO/ISPF INITIALIZATION
+FEK900I ISPF IVP: ( TSO/ISPF session will be initialized )
+FEK900I ISPF IVP: RC=0
+FEK900I ISPF IVP: MSG: SUCCESSFUL
+FEK900I ISPF IVP: -----
-----
+FEK900I ISPF IVP: *** CHECK: Shutting down TSO/ISPF IVP session
+FEK900I ISPF IVP: RC=0
+FEK900I ISPF IVP: MSG: SUCCESSFUL
+FEK900I ISPF IVP: -----
-----
+FEK900I ISPF IVP: Host installation verification completed successfully
+FEK900I ISPF IVP: -----
-----
+FEK901I ISPF IVP Exit code = 0

```

注:

- この機能は、fekfivpi IVP (インストール検査プログラム) の機能に似ています。
- RSE デーモンにより、IVP のパスワードとして使用する PassTicket が生成されるため、パスワードを要求する要応答オペレーター宛メッセージ (WTOR) はありません。

IVP PASSTICKET,userid

ユーザー ID userid 用に生成された PassTicket の再利用性をテストします。結果は 1 つ以上の FEK900I コンソール・メッセージとともに表示されます。戻りコードは、コンソール・メッセージ FEK901I とともに表示されます。

```

+FEK900I PASSTICKET IVP: the default applid=FEKAPPL
+FEK900I PASSTICKET IVP: Success, PassTicket IVP finished normally
+FEK901I PASSTICKET IVP Exit code = 0

```

注:

- RACF をセキュリティ製品として使用する場合、再使用可能な PassTickets は、セキュリティ定義に「NO REPLAY PROTECTION」キーワードが必要です。
- このテストを行うための、同等の IVP (インストール検査プログラム) はありません。IVP=IVP 引数を使用して RSE デーモンを開始すると、PassTicket 生成をテストする PassTicket IVP が呼び出されますが、これは PassTicket の再利用性をテストすることができません。
- RSE デーモンにより、IVP のパスワードとして使用する PassTicket が生成されるため、パスワードを要求する要応答オペレーター宛メッセージ (WTOR) はありません。

SWITCH

新しい監査ログ・ファイルに切り替えます。

- 変数はイタリック体の英小文字で表記され、ユーザーが指定する名前または値を表します。例えば、データ・セット名は変数です。変数は、大/小文字が区別される場合があります。

構文例

次の例では、USER コマンドはキーワードです。必須の変数パラメーターは `user_id` で、オプションの変数パラメーターは `password` です。これらの変数パラメーターは、ユーザー独自の値に置き換えてください。

```
>>—USER—user_id—————><
      |—————|
      |password—|
```

非英数字およびブランク・スペース

小括弧、ピリオド、コンマ、等号、ブランク・スペースなど、構文図に英数字以外の文字が示されている場合は、その文字を構文の一部としてコーディングする必要があります。この例では、`OPERAND=(001 0.001)` とコーディングする必要があります。

```
>>—OPERAND—=—(—001— —0.001—)—————><
```

複数のオペランドの選択

オペランド・グループ内で左に戻る矢印は、複数のオペランドを選択できるか、単一のオペランドを反復できることを意味しています。

```
>>—————><
      |—————|
      |REPEATABLE_OPERAND_1—|
      |REPEATABLE_OPERAND_2—|
      |—————|
      |—————|
```

1 行より長い場合

図が 1 行より長い場合、最初の行は単一の矢印で終わり、2 行目は単一の矢印で始まります。

```
>>—| The first line of a syntax diagram that is longer than one line |—>
>—| The continuation of the subcommands, parameters, or both |—————><
```

構文フラグメント

一部の図には、構文フラグメントが含まれている場合があります。これは、長すぎたり、複雑すぎたり、繰り返しが多すぎる構文を分割するためのものです。構文フラグメント名は大/小文字混合で表記され、構文図の中とフラグメントの見出しに表示されます。フラグメントは、メインの図の下に配置されます。

```
>>—| Syntax fragment |—————><
```

```
Syntax fragment:
|—1ST_OPERAND—,—2ND_OPERAND—,—3RD_OPERAND—|
```

ホスト構成リファレンス

このセクションでは、「*IBM Rational Developer for System z* ホスト構成リファレンス」(SA88-4226) に記載されている情報を要約しています。詳細については、この資料を参照してください。

Developer for System z について

Developer for System z ホスト・システムは、クライアントがホスト・システム・サービスとデータにアクセスできるようにするために相互に作用する、複数のコンポーネントで構成されています。これらのコンポーネントの設計を理解しておく、構成に関して適切な判断を行うことができます。

セキュリティに関する考慮事項

Developer for System z では、メインフレーム以外のワークステーション上にいるユーザーがメインフレームにアクセスできます。このため、接続要求の妥当性検査、ホスト・システムとワークステーション間のセキュアな通信の提供、およびアクティビティの許可と監査が、製品構成の重要な側面となります。

TCP/IP に関する考慮事項

Developer for System z では、TCP/IP を使用して、非メインフレーム・ワークステーションのユーザーに、メインフレームからアクセスすることができます。また、さまざまなコンポーネントやその他の製品同士の間での通信にも TCP/IP を使用します。

WLM に関する考慮事項

従来の z/OS アプリケーションとは異なり、Developer for System z は、ワークロード・マネージャー (WLM) で容易に識別できる一体構造のアプリケーションではありません。Developer for System z は、クライアントがホスト・システムのサービスとデータにアクセスできるようにするために相互に作用する、複数のコンポーネントで構成されています。これらのサービスの一部は異なるアドレス・スペースでアクティブとなるため、WLM 分類も異なることになります。

チューニングに関する考慮事項

RSE (リモート・システム・エクスプローラー) は、Developer for System z の中核です。クライアントからの接続とワークロードを管理するために、RSE は、スレッド・プーリング・アドレス・スペースを制御するデーモン・アドレス・スペースから構成されています。このデーモンは、接続と管理の目的のためのフォーカル・ポイントとして機能し、それに対してスレッド・プールは、クライアントのワークロードを処理します。

この構成により、Developer for System z セットアップをチューニングする場合の主要な対象は RSE となります。ただし、それぞれが 17 個以上のスレッドを使用する何百人ものユーザー、ある程度の大きさのストレージ、そして場合によっては 1 つ以上のアドレス・スペースを保守するには、Developer for System z と z/OS の両方を適切に構成する必要があります。

パフォーマンスに関する考慮事項

z/OS は高度にカスタマイズ可能なオペレーティング・システムであり、システムの場合によっては小さな変更で全体のパフォーマンスに大きな影響を与えることができます。この章では、Developer for System z のパフォーマンスを向上させるために行うことができるいくつかの変更を中心に説明します。

クライアントへのプッシュ機能に関する考慮事項

クライアントへのプッシュ、またはホスト・ベースのクライアント制御は、次の中央管理をサポートしています。

- クライアント構成ファイル
- クライアント製品バージョン
- プロジェクト定義

CICSTS に関する考慮事項

この章には、CICS Transaction Server 管理者に有益な情報が記載されています。

ユーザー出口に関する考慮事項

本章は、出口ルーチンを作成することによって Developer for System z を拡張する上で役立ちます。

TSO 環境のカスタマイズ

この章は、Developer for System z で TSO 環境に DD ステートメントとデータ・セットを追加することにより、TSO ログオン・プロシーチャーを模倣するのに役立ちます。

複数のインスタンスの実行

同じシステム上で Developer for System z の複数のインスタンスをアクティブにしたい場合があります。例えば、アップグレードをテストするときなどです。しかし、TCP/IP ポートなど、一部のリソースは共用できないため、デフォルトが常に適用可能であるとは限りません。この章の情報をを使用して Developer for System z のさまざまなインスタンスの共存を計画してください。その後、この構成ガイドを使用して、それらのインスタンスをカスタマイズすることができます。

構成問題のトラブルシューティング

この章は、Developer for System z の構成時に起きる可能性があるいくつかの一般的な問題について、ユーザーを支援するためのもので、以下のセクションで構成されています。

- ログとセットアップの分析に使用、FEKLOGS
- ログ・ファイル
- ダンプ・ファイル
- トレース
- z/OS UNIX 許可ビット
- 予約済み TCP/IP ポート
- アドレス・スペース・サイズ
- APPC トランザクションおよび TSO コマンド・サービス
- 各種情報

SSL および X.509 認証のセットアップ

このセクションは、Secure Socket Layer (SSL) のセットアップ時、または既存のセットアップの検査時や変更時に起きる可能性があるいくつかの一般的な問題について、ユーザーを支援するためのものです。また、このセクションには、X.509 証明書で自分自身を認証するユーザーをサポートするためのサンプルのセットアップも記載されています。

AT-TLS のセットアップ

このセクションは、Application Transparent Transport Layer Security (AT-TLS) のセットアップ時、または既存のセットアップの検査時や変更時に起きる可能性があるいくつかの一般的な問題について、ユーザーを支援するためのものです。

TCP/IP のセットアップ

このセクションは、TCP/IP のセットアップ時、または既存のセットアップの検査時や変更時に起きる可能性があるいくつかの一般的な問題について、ユーザーを支援するためのものです。

参考文献

参考資料

本書では、以下の資料を参照しています。

表 30. 参考資料

資料名	資料番号	参照	参照 Web サイト
IBM Rational Developer for System z Program Directory	GI88-4172	Developer for System z	http://www-01.ibm.com/support/docview.wss?uid=swg27038517
Program Directory for IBM Rational Developer for System z Host Utilities	GI88-4326	Developer for System z	http://www-01.ibm.com/support/docview.wss?uid=swg27038517
IBM Rational Developer for System z前提条件	SC88-4704	Developer for System z	http://www-01.ibm.com/support/docview.wss?uid=swg27038517
IBM Rational Developer for System zホスト構成クイック・スタート	GI88-4171	Developer for System z	http://www-01.ibm.com/support/docview.wss?uid=swg27038517
IBM Rational Developer for System z ホスト構成ガイド	SC88-5663	Developer for System z	http://www-01.ibm.com/support/docview.wss?uid=swg27038517
IBM Rational Developer for System zホスト構成リファレンス	SA88-4226	Developer for System z	http://www-01.ibm.com/support/docview.wss?uid=swg27038517
IBM Rational Developer for System z ホスト構成ユーティリティー・ガイド	SA88-4197	Developer for System z	http://www-01.ibm.com/support/docview.wss?uid=swg27038517
IBM Rational Developer for System z メッセージおよびコード	SA88-4565	Developer for System z	http://www-01.ibm.com/support/docview.wss?uid=swg27038517
IBM Rational Developer for System z共通ホスト構成と保守に関する問題への回答	SA88-5113	Developer for System z	http://www-01.ibm.com/support/docview.wss?uid=swg27038517
IBM Rational Developer for System z Common Access Repository Manager Developer's Guide	SC23-7660	Developer for System z	http://www-01.ibm.com/support/docview.wss?uid=swg27038517
IBM Rational Developer for System z前提条件	SC88-4704	Developer for System z	http://www.ibm.com/software/rational/products/developer/systemz/library/index.html
IBM Rational Developer for System zホスト構成クイック・スタート	GI88-4171	Developer for System z	http://www.ibm.com/software/rational/products/developer/systemz/library/index.html
SCLM Developer Toolkit 管理者ガイド	SC88-5664	Developer for System z	http://www-01.ibm.com/support/docview.wss?uid=swg27038517
Using APPC to provide TSO command services	SC14-7291	ホワイト・ペーパー	http://www-01.ibm.com/support/docview.wss?uid=swg27038517

表 30. 参考資料 (続き)

資料名	資料番号	参照	参照 Web サイト
Using ISPF Client Gateway to provide CARMA services	SC14-7292	ホワイト・ペーパー	http://www-01.ibm.com/support/docview.wss?uid=swg27038517
Communications Server IP 構成ガイド	SC88-8926	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
Communications Server IP 構成解説書	SC88-8927	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
Communications Server IP Diagnosis Guide	GC31-8782	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
Communications Server IP システム管理者のコマンド	SC88-9073	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
Communications Server SNA ネットワーク・インプリメンテーション・ガイド	SC88-8928	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
Communications Server SNA オペレーション	SC88-8930	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
Cryptographic Services System SSL (Secure Sockets Layer) プログラミング	SD88-6252	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
DFSMS Macro Instructions for Data Sets	SC26-7408	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
DFSMS データ・セットの使用法	SC88-9114	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
言語環境プログラム カスタマイズ	SA88-8552	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
言語環境プログラム デバッグ・ガイド	GA88-8548	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
MVS 診断: ツールと保守援助プログラム	GA88-8561	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
MVS 初期設定およびチューニング ガイド	SA88-8563	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
MVS 初期設定およびチューニング解説書	SA88-8564	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
MVS JCL 解説書	SA88-8569	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
MVS 計画 APPC/MVS 管理	SA88-8571	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
MVS 計画: ワークロード管理	SA88-8574	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
MVS システム・コマンド	SA88-8593	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
Security Server RACF コマンド言語解説書	SA88-8617	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
Security Server RACF セキュリティー管理者のガイド	SA88-8613	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
TSO/E カスタマイズ	SA88-8629	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
TSO/E REXX 解説書	SA88-8635	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/

表 30. 参考資料 (続き)

資料名	資料番号	参照	参照 Web サイト
UNIX System Services コマンド解説書	SA88-8641	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
UNIX System Services 計画	GA88-8639	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
UNIX システム・サービス ユーザーズ・ガイド	SA88-8640	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
REXX および z/OS UNIX システム・サービスの使い方	SA88-8644	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
Java Diagnostic Guide	SC34-6650	Java 6.0	http://www.ibm.com/developerworks/java/jdk/diagnosis/
Java SDK and Runtime Environment User Guide	/	Java 6.0	http://www-03.ibm.com/servers/eserver/zseries/software/java/
Resource Definition Guide	SC34-6430	CICSTS 3.1	http://www-03.ibm.com/systems/z/os/zos/bkserv/zapplsbooks.html
Resource Definition Guide	SC34-6815	CICSTS 3.2	http://www-03.ibm.com/systems/z/os/zos/bkserv/zapplsbooks.html
Resource Definition Guide	SC34-7000	CICSTS 4.1	https://publib.boulder.ibm.com/infocenter/cicsts/v4r1/index.jsp?topic=/com.ibm.cics.ts.home.doc/library/library_html.html
Resource Definition Guide	SC34-7181	CICSTS 4.2	https://publib.boulder.ibm.com/infocenter/cicsts/v4r2/index.jsp?topic=/com.ibm.cics.ts.home.doc/library/library_html.html
RACF Security Guide	SC34-6454	CICSTS 3.1	http://www-03.ibm.com/systems/z/os/zos/bkserv/zapplsbooks.html
RACF Security Guide	SC34-6835	CICSTS 3.2	http://www-03.ibm.com/systems/z/os/zos/bkserv/zapplsbooks.html
RACF Security Guide	SC34-7003	CICSTS 4.1	https://publib.boulder.ibm.com/infocenter/cicsts/v4r1/index.jsp?topic=/com.ibm.cics.ts.home.doc/library/library_html.html
RACF Security Guide	SC34-7179	CICSTS 4.2	https://publib.boulder.ibm.com/infocenter/cicsts/v4r2/index.jsp?topic=/com.ibm.cics.ts.home.doc/library/library_html.html
言語解説書	SC88-9117	Enterprise COBOL for z/OS	http://www-03.ibm.com/systems/z/os/zos/bkserv/zapplsbooks.html

本書では、以下の Web サイトを参照しています。

表 31. 参照される Web サイト

説明	参照 Web サイト
Developer for System z IBM Knowledge Center	http://www-01.ibm.com/support/knowledgecenter/SSQ2R2_9.1.0/com.ibm.etools.getstart.wsentdev.doc/kc_version_welcome_rdz.html
Developer for System z ライブラリー	http://www-01.ibm.com/support/docview.wss?uid=swg27038517
Developer for System z ホーム・ページ	http://www-03.ibm.com/software/products/en/developerforsystemz/
Developer for System z 推奨サービス	http://www-01.ibm.com/support/docview.wss?rs=2294&context=SS2QJ2&uid=swg27006335

表 31. 参照される Web サイト (続き)

説明	参照 Web サイト
Developer for System z 機能拡張依頼	https://www.ibm.com/developerworks/support/rational/rfe/
z/OS インターネット・ライブラリー	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
CICS TS IBM Knowledge Center	https://publib.boulder.ibm.com/infocenter/cicsts/v4r1/index.jsp
IBM Tivoli® Directory Server	http://www-01.ibm.com/software/tivoli/products/directory-server/
Problem Determination Tools Plug-ins	http://www-01.ibm.com/software/awdtools/deployment/pdtpplugins/
Java セキュリティー情報	http://www.ibm.com/developerworks/java/jdk/security/
Apache Ant のダウンロード	http://ant.apache.org/
Java keytool の資料	http://java.sun.com/j2se/1.5.0/docs/tooldocs/solaris/keytool.html
CA サポート・ホーム・ページ	https://support.ca.com/

情報資料

以下の資料は、必要なホスト・システム・コンポーネントのセットアップの問題を理解するのに役立ちます。

表 32. 情報資料

資料名	資料番号	参照	参照 Web サイト
ABCs of z/OS System Programming Volume 9 (z/OS UNIX)	SG24-6989	Redbook	http://www.redbooks.ibm.com/
System Programmer's Guide to: Workload Manager	SG24-6472	Redbook	http://www.redbooks.ibm.com/
TCPIP Implementation Volume 1: Base Functions, Connectivity, and Routing	SG24-7532	Redbook	http://www.redbooks.ibm.com/
TCPIP Implementation Volume 3: High Availability, Scalability, and Performance	SG24-7534	Redbook	http://www.redbooks.ibm.com/
TCP/IP Implementation Volume 4: Security and Policy-Based Networking	SG24-7535	Redbook	http://www.redbooks.ibm.com/
Tivoli Directory Server for z/OS	SG24-7849	Redbook	http://www.redbooks.ibm.com/

特記事項

© Copyright IBM Corporation 1992, 2013.

本書は米国 IBM が提供する製品およびサービスについて作成したものです。

本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒103-8510

東京都中央区日本橋箱崎町19番21号

日本アイ・ビー・エム株式会社

法務・知的財産

知的財産権ライセンス渉外

以下の保証は、国または地域の法律に沿わない場合は、適用されません。IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

Intellectual Property Dept. for Rational Software
IBM Corporation
Silicon Valley Lab
555 Bailey Avenue
San Jose, CA 95141-1003
U.S.A.

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性がありますが、その測定値が、一般に利用可能なシステムのものと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確証できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者をお願いします。

IBM の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があります、単に目標を示しているものです。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

著作権使用許諾

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを

経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほのめかしたり、保証することはできません。サンプル・プログラムは、現存するままの状態を提供され、いかなる保証条件も適用されません。IBM は、お客様の当該サンプル・プログラムの使用から生ずるいかなる損害に対しても一切の責任を負いません。

それぞれの複製物、サンプル・プログラムのいかなる部分、またはすべての派生的創作物にも、次のように、著作権表示を入れていただく必要があります。

© (お客様の会社名) (西暦年). このコードの一部は、IBM Corp. のサンプル・プログラムから取られています。© Copyright IBM Corp. 1992, 2013.

この情報をソフトコピーでご覧になっている場合は、写真やカラーの図表は表示されない場合があります。

プライバシー・ポリシーに関する考慮事項

サービス・ソリューションとしてのソフトウェアも含めた IBM ソフトウェア製品 (「ソフトウェア・オファリング」) では、製品の使用に関する情報の収集、エンド・ユーザーの使用感の向上、エンド・ユーザーとの対話またはその他の目的のために、Cookie はじめさまざまなテクノロジーを使用することがあります。多くの場合、ソフトウェア・オファリングにより個人情報が収集されることはありません。IBM の「ソフトウェア・オファリング」の一部には、個人情報を収集できる機能を持つものがあります。ご使用の「ソフトウェア・オファリング」が、これらの Cookie およびそれに類するテクノロジーを通じてお客様による個人情報の収集を可能にする場合、以下の具体的事項を確認ください。

この「ソフトウェア・オファリング」は、Cookie もしくはその他のテクノロジーを使用して個人情報を収集することはありません。

商標

IBM、IBM ロゴおよび ibm.com は、世界の多くの国で登録された International Business Machines Corp. の商標です。他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。現時点での IBM の商標リストについては、www.ibm.com/legal/copytrade.shtml をご覧ください。

製品資料に関する使用条件

適用範囲

IBM Web サイトのご利用条件に加えて、以下のご使用条件があります。

個人使用

これらの資料は、すべての著作権表示その他の所有権表示をしていただくことを条件に、非商業的な個人による使用目的に限り複製することができます。ただし、IBM の明示的な承諾をえずに、これらの資料またはその一部について、二次的著作物を作成したり、配布 (頒布、送信を含む) または表示 (上映を含む) することはできません。

商業的使用

これらの資料は、すべての著作権表示その他の所有権表示をしていただくことを条件に、お客様の企業内に限り、複製、配布、および表示することができます。ただし、IBM の明示的な承諾をえずにこれらの資料の二次的著作物を作成したり、お客様の企業外で資料またはその一部を複製、配布、または表示することはできません。

権利

ここで明示的に許可されているもの以外に、資料や資料内に含まれる情報、データ、ソフトウェア、またはその他の知的所有権に対するいかなる許可、ライセンス、または権利を明示的にも黙示的にも付与するものではありません。

資料の使用が IBM の利益を損なうと判断された場合や、上記の条件が適切に守られていないと判断された場合、IBM はいつでも自らの判断により、ここで与えた許可を撤回できるものとさせていただきます。

お客様がこの情報をダウンロード、輸出、または再輸出する際には、米国のすべての輸出入関連法規を含む、すべての関連法規を遵守するものとします。

IBM は、これらの資料の内容についていかなる保証もしません。これらの資料は、特定物として現存するままの状態を提供され、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任なしで提供されます。

著作権使用許諾

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほのめかしたり、保証することはできません。サンプル・プログラムは、現存するままの状態を提供され、いかなる保証条件も適用されません。IBM は、お客様の当該サンプル・プログラムの使用から生ずるいかなる損害に対しても一切の責任を負いません。

商標の帰属表示

IBM、IBM ロゴおよび ibm.com は、世界の多くの国で登録された International Business Machines Corp. の商標です。他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。現時点での IBM の商標リストについては、www.ibm.com/legal/copytrade.shtml をご覧ください。

Adobe および PostScript は、Adobe Systems Incorporated の米国およびその他の国における登録商標または商標です。

Cell Broadband Engine は、Sony Computer Entertainment, Inc. の米国およびその他の国における商標であり、同社の許諾を受けて使用しています。

Intel、Intel Centrino、Intel SpeedStep、Intel Xeon、Celeron、Itanium、および Pentium は、Intel Corporation または子会社の米国およびその他の国における商標または登録商標です。

IT Infrastructure Library は英国 Office of Government Commerce の一部である the Central Computer and Telecommunications Agency の登録商標です。

ITIL は英国 The Minister for the Cabinet Office の登録商標および共同体登録商標であって、米国特許商標庁にて登録されています。

Linear Tape-Open、LTO、および Ultrium は、HP、IBM Corp. および Quantum の米国およびその他の国における商標です。

Linux は、Linus Torvalds の米国およびその他の国における登録商標です。

Microsoft、Windows、および Windows ロゴは、Microsoft Corporation の米国およびその他の国における商標です。

Java およびすべての Java 関連の商標およびロゴは Oracle やその関連会社の米国およびその他の国における商標です。

UNIX は The Open Group の米国およびその他の国における登録商標です。



Printed in Japan

SC88-5663-13



日本アイ・ビー・エム株式会社
〒103-8510 東京都中央区日本橋箱崎町19-21